

Abre tu nevera!

Aplicación para Android

Memoria del proyecto

Autora:	Rosa Cañizares Fernández
Fecha:	27/06/2014
Director:	LuísPerez Vidal
Departamento:	LSI
Titulación:	Ingeniería informática
Centre:	Facultatd'Informàtica de Barcelona (FIB)
Universitat:	UniversitatPolitécnica de Catalunya (UPC)
	BarcelonaTech

Agradecimientos

En primer lugar, me gustaría agradecer a mi tutor, Luís Perez Vidal, que aceptara la idea que le propuse para la realización de este proyecto. Gracias a ello pude empezar el proyecto muy motivada

También me gustaría agradecer a todos los profesores que he tenido durante la carrera y a los compañeros que me han acompañado durante todo este largo camino y que han hecho de mi vida universitaria toda una experiencia que me va a marcar durante toda la vida.

A continuación dar las gracias a mis amigos y compañeros de trabajo que día a día me han apoyado y me han animado para poder tirar este proyecto adelante. Sin vuestras palabras de ánimo y vuestra “pesadez” preguntandome sobre el proyecto, acabar esto habría sido imposible.

También debo dar las gracias a todas las personas que han colaborado en la creación de la aplicación, desde mis testadores hasta mi “asistente de diseño”. Gracias a vosotros, la aplicación ha quedado así de bien.

Y por último, debo agradecer a mis padres. Por animarme cada día y no dejar que me viniera abajo a cada momento de debilidad. Por estar ahí en cada paso de mi vida. Gracias a vosotros, soy mucho mejor persona.

Tabla de contenidos

1. Introducción	5
1.1 Motivación.....	5
1.2 Descripción de los objetivos.....	6
1.3 Organización de la memoria	7
2. Android	9
2.1 Historia de Android	9
2.2 Características	10
2.3 Arquitectura.....	11
2.4 Versiones de Android y cuota de mercado	12
2.5 Aplicaciones en Android	15
2.5.1 Componentes de una aplicación.....	15
2.5.2 Estructura de un proyecto Android.....	18
3. Cocina conmigo y otras aplicaciones.....	22
3.1 Cocina conmigo!.....	22
3.2 App Canal cocina	23
3.3 EpicuriousApp	24
4. Análisis.....	25
4.1 Definición del proyecto.....	25
4.2 Análisis de requisitos	26
4.2.1 Requisitos funcionales	26
4.2.2 Requisitos no funcionales	27
4.3 Casos de uso	28
4.3.1 Diagrama de casos de uso	28
4.3.2 Casos de uso:.....	29
CU1: Buscar receta:	29
CU2: Ver receta.....	30
CU3: Lanzar reloj de cocina:.....	32
CU4: Visualizar listas de la compra.....	33
CU5: Gestionar lista de la compra	33
CU6: Visualizar mi nevera.....	35
CU7: Configurar alarma de reloj de cocina	36
4.4 Modelo conceptual.....	37
5. Diseño e implementación de la aplicación.....	38
5.1 Diseño de la base de datos y SQLite	38
5.2 Prototipos y diagrama de navegación	42
5.3 Herramientas utilizadas durante el proyecto	47
6. Fase de test, depuración de errores y problemas encontrados durante el desarrollo.	50
6.1 Pruebas en emulador	50
6.2 Genymotion y su incorporación a Eclipse.....	51
6.3 Pruebas en móviles reales.....	53
7. Planificación del proyecto	56
8. Gestión de costes	61
9. Posibles ampliaciones	64
10. Manual de usuario.....	66

11. Conclusiones	83
11.1 Evaluación de objetivos	83
11.2 Conclusiones personales	84
11. Bibliografía	85

1. Introducción

En este primer capítulo de la memoria, se explicará cuales han sido los motivos que me han llevado a realizar este proyecto, tanto las personales como las profesionales.

Además de explicar estos motivos, también se detallarán los objetivos que se pretenden cumplir y dar un vistazo a la organización de la memoria para que a simple vista, el lector pueda ir de una forma más rápida al apartado que le pueda interesar.

1.1 Motivación

Desde hace varios años (primero como personal en prácticas y, más tarde, como personal en plantilla), trabajo como desarrolladora de aplicaciones web. Personalmente, es un trabajo que me gusta mucho, pero es una actividad que desarrollo a diario durante muchas horas al mes. Por lo tanto, desde que fui consciente de que tenía que realizar el proyecto de final de carrera en la universidad, supe que quería hacer algo que fuera bastante distinto, pero que no se alejara demasiado de lo que es el desarrollo de una aplicación.

Debido a que el desarrollo de aplicaciones móviles es algo que está al alza y que hay una gran comunidad de desarrolladores que facilitan el aprendizaje sea cual sea la plataforma, me decidí a realizar una aplicación para un dispositivo móvil.

Una vez decidido que iba a realizar una aplicación para móvil, fue fácil decidir la plataforma, ya que Android, al ser código abierto y compatible con muchos dispositivos hacía mi trabajo más fácilmente utilizable. Por otra parte, no necesitaba ninguna licencia especial para poder desarrollar aplicaciones bajo esa plataforma.

En cuanto a la temática escogida, al ser un proyecto al que se le tiene que dedicar mucho tiempo, tuve claro que tenía que ser algo que me gustara y puesto que soy una gran aficionada a la cocina, la decisión fue fácil.

.

La idea en sí era algo que se me había ocurrido hace bastante tiempo, al darme cuenta que en casa era bastante frecuente improvisar una comida contando con “lo que había en la nevera”. Por eso pensé que había que buscar una manera de automatizar este aparente caos.

Finalmente gracias a la inspiración en la capacidad de mi madre para improvisar una comida, al auge de los programas de cocina de la televisión y a esta oportunidad de realizar una aplicación en Android, nació: Abre tu nevera!

1.2 Descripción de los objetivos

El principal objetivo de este proyecto es crear una aplicación para Android que cualquier persona, independientemente de su edad o de su experiencia tecnológica pueda utilizar fácilmente.

Para lograrlo, hay que cumplir una serie de subobjetivos con el fin de poder conseguir el objetivo final:

- *Estudiar y aprender cómo funciona Android:* Parto de un desconocimiento total sobre el desarrollo en plataformas Android, además de que mi experiencia sobre Java ha caído en desuso durante este tiempo en el que me he dedicado a otros lenguajes. Por tanto, lo primero fue aprender el funcionamiento de este entorno y refrescar los conocimientos sobre el citado lenguaje de programación.
- *Analizar, diseñar e implementar una aplicación desde cero:* Debido a que no es una aplicación basada en ninguna otra existente, sino que es una idea más o menos original, a parte del desarrollo de la aplicación, debo realizar primero una pequeña investigación en el mercado para ver que otras aplicaciones hay para que la mía tenga algún punto diferenciador que pueda hacer que destaque, además de pensar en todas las características que pueda tener. Una vez pensadas, hay que trasladarlas al diseño e implementarlas para poder obtener la aplicación final.

- *Revisar y comprobar que el funcionamiento de la aplicación satisface los requisitos pensados inicialmente:* Una vez finalizada la implementación, se tiene que revisar que todos los conceptos pensados inicialmente estén incluidos dentro de la aplicación, y en el caso en que no lo estén ver el motivo por el cual no están.

1.3 Organización de la memoria

Para tener una vista global rápida de esta memoria se facilita un pequeño resumen de cada uno de los puntos. Ésta memoria cuenta en total con 11 capítulos.

En el primer capítulo, como ya hemos visto, se explican las motivaciones del proyecto y los objetivos que se desean obtener una vez finalizado.

En el segundo capítulo (2. Android) veremos que es en sí Android, que características tiene, cual es su arquitectura y haremos un pequeño repaso a su historia.

En el tercer capítulo (3. Cocina conmigo y otras aplicaciones) podremos ver productos similares a *Abre tu nevera* que existen actualmente en el mercado. Veremos tanto aplicaciones para dispositivos móviles como productos que encontramos en otras plataformas.

En el cuarto capítulo (4. Análisis) entraremos en profundidad en la definición y especificación del proyecto. Veremos los requisitos de la aplicación y sus casos de uso. Una vez leído este capítulo, el lector debería tener una idea bastante clara de lo que se pretende realizar en este proyecto y que funcionalidades tendrá la aplicación. En el quinto capítulo (5. Diseño e implementación de la aplicación.), podremos ver cómo funciona internamente la aplicación. Veremos los diagramas de secuencia creados a partir de los casos de uso. También podremos ver en este capítulo el storyboard de la aplicación a partir de los prototipos creados y qué herramientas se han usado durante la realización del proyecto.

En el sexto capítulo (6. Fase de test, depuración de errores y problemas encontrados durante el desarrollo.), veremos que herramientas hemos usado para realizar los

tests, los resultados de estos, que problemas nos hemos encontrado y como los hemos resuelto.

En el séptimo capítulo (7. Planificación del proyecto) veremos la planificación realizada al inicio del proyecto y como se ha ido modificando hasta llegar a la planificación final.

En el octavo capítulo (8. Gestión de costes), veremos lo que costaría la realización del proyecto en un entorno real basado en las horas calculadas en el anterior capítulo.

En el noveno capítulo (9. Posibles mejoras) veremos cómo podríamos mejorar el producto final mediante una serie de ampliaciones que la harían más competitiva si fuera puesta a la venta en el mercado real.

En el décimo capítulo (10. Manual de usuario), veremos el manual de usuario de la aplicación para que cualquier persona que lea la memoria, pueda usar la aplicación fácilmente y sin necesidad de ayuda.

Por último, en el undécimo capítulo, (11. Conclusiones) veremos las conclusiones obtenidas del desarrollo de este proyecto

2. Android

Como el proyecto está desarrollado para la plataforma Android, antes de meternos de lleno en lo que es la aplicación en sí mediante la especificación, diseño y desarrollo, veremos primero una pequeña introducción a qué es Android, tanto de donde procede y como se creó, a una pequeña introducción a su arquitectura y la manera en la que se puede desarrollar para esta plataforma.

2.1 Historia de Android

Android es tanto un sistema operativo como una plataforma de desarrollo para dispositivos móviles tales como teléfonos móviles o tablets. Está basado en el kernel de Linux.

Inicialmente, fue desarrollado por Android, Inc con la ayuda económica de Google, que, poco después, fue comprada en 2005 al ingeniero de software Andi Rubin con la intención de crear un sistema operativo para móvil, ya que era un mercado que estaba empezando a despertar.

Android fue presentado por primera vez en 2007 junto a la fundación Open Handset Alliance, un conglomerado de compañías de software, telecomunicaciones, hardware y operadores de servicio. A la misma vez que fue presentada la OHA, fue presentado el Android Software DevelopmentKit (SDK). A finales de septiembre del siguiente año, después de evaluar la estabilidad de su primera beta, lanzaron la primera versión oficial de Android (Android 1.0 SDK)

El primer dispositivo móvil con sistema operativo Android fue el HTC Dream y se puso a la venta en octubre de 2008.



Imagen 1 HTC Dream

Actualmente, Android cuenta con una gran comunidad de desarrolladores creando aplicaciones para esta plataforma. Esto es debido a que, por una parte, el desarrollo de aplicaciones es gratuito y, por otro lado, su lenguaje de desarrollo está basado en java, un lenguaje con una alta penetración en el mercado.

Existen ya más de un millón de aplicaciones desarrolladas para Android, de las cuales dos tercios son gratuitas. Estas cifras no tienen en cuenta las tiendas no oficiales en donde también se pueden obtener aplicaciones para Android, como la tienda de aplicaciones de Samsung o la Amazon Appstore.

2.2 Características

Entre las características principales de Android, podemos destacar:

Diseño: La plataforma es adaptable a pantallas de gran resolución además de obtener gráficos con OpenGL ES 2.0

Almacenamiento: Android dispone de una base de datos llamada SQLite muy ligera que se puede integrar en cualquier aplicación

Navegador web: Dispone de un navegador web basado en el motor de renderizado WebKit

Soporte multimedia: Android soporta diferentes formatos tanto de imagen como de sonido y vídeo tales como WebM, 3GP, MP4, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, MP3, MIDI, OggVorbis, WAV, JPEG, PNG, GIF y BMP.

Entorno de desarrollo: Por defecto, incluye un emulador integrado dentro de Eclipse que se puede utilizar usando el plugin de Herramientas de Desarrollo de Android.

2.3 Arquitectura

Antes de poder empezar a desarrollar en Android es importante conocer como está estructurado y de qué manera funciona internamente. Para eso procederemos a ver que compone la arquitectura de Android y cómo funcionan cada una de sus diferentes capas:

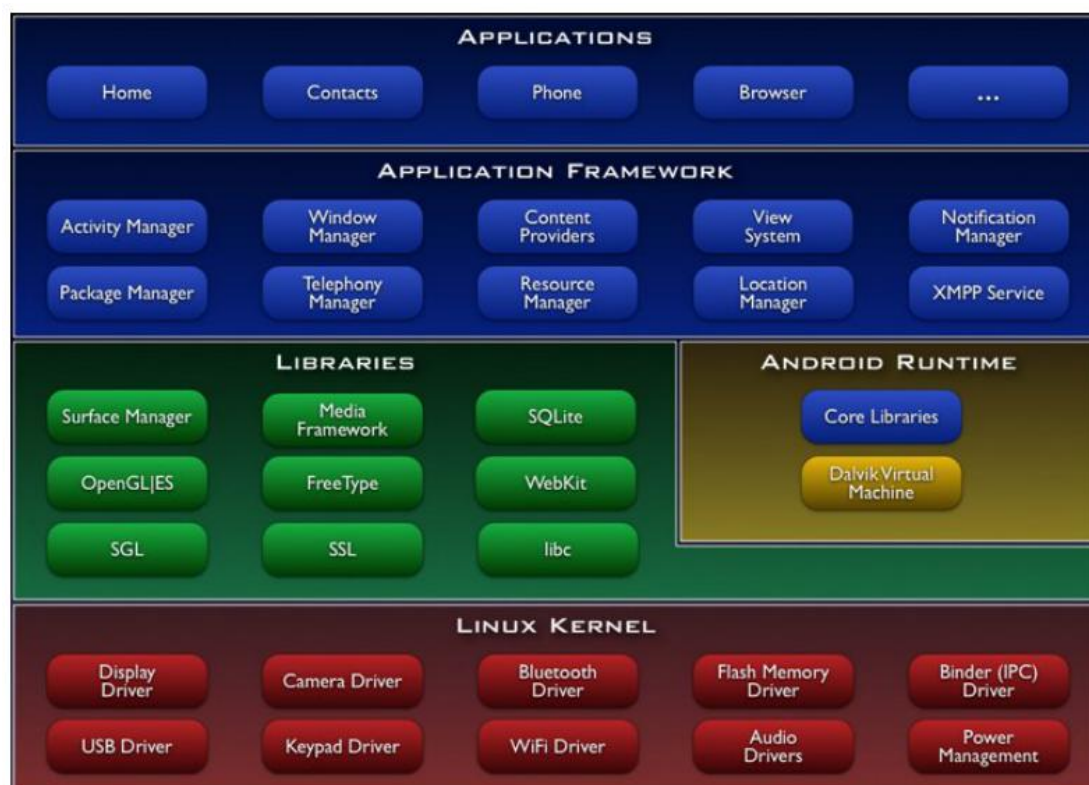


Imagen 2: Arquitectura de Android

- *Aplicaciones:* Capa formada por todas las aplicaciones del dispositivo, tanto si tienen interfaz de usuario como si no la tienen. Las aplicaciones que se van instalando en el dispositivo, se van añadiendo también.

- *Framework de las aplicaciones:* Desde aquí, los desarrolladores pueden acceder a las mismas aplicaciones que se usan dentro de las aplicaciones base que están dentro del dispositivo como por ejemplo la cámara.
- *Librerías:* También se incluye un conjunto de librerías C/C++ usadas por varios componentes del sistema. Dentro de estas librerías podemos encontrar OpenGL ES o SQLite.
- *Runtime de Android:* Se trata de un conjunto de bibliotecas que proporciona la mayoría de funcionalidades disponibles en las librerías básicas de Java, así como otras específicas de Android.
- *Núcleo de Linux:* Ya que Android está basada en el kernel de Linux, desde aquí se gestionan servicios básicos como la seguridad, la memoria, la pantalla o la gestión de procesos.

2.4 Versiones de Android y cuota de mercado

Desde que Android lanzó su primera versión comercial en 2008, se han ido lanzando nuevas versiones que mejoran algunas características e incluyen otras nuevas.

Cada una de las versiones de Android tienen un nombre en clave de un postre o de un dulce y la primera letra de ese postre, sigue un orden alfabético.

A continuación tenemos las estadísticas de la distribución de versiones dentro de los dispositivos actuales (datos de febrero de 2014):

1. Android 4.1.x JellyBean -> 35.5%
2. Android 2.3.x Gingerbread ->20%
3. Android 4.2.x JellyBean -> 16.3%
4. Android 4.0.x Ice CreamSandwich -> 16.1%
5. Android 4.3 JellyBean -> 8.9%
6. Android 4.4 KitKat -> 1.8%
7. Android 2.2 Froyo -> 1.3%
8. Android 3.2 Honeycomb -> 0.1%

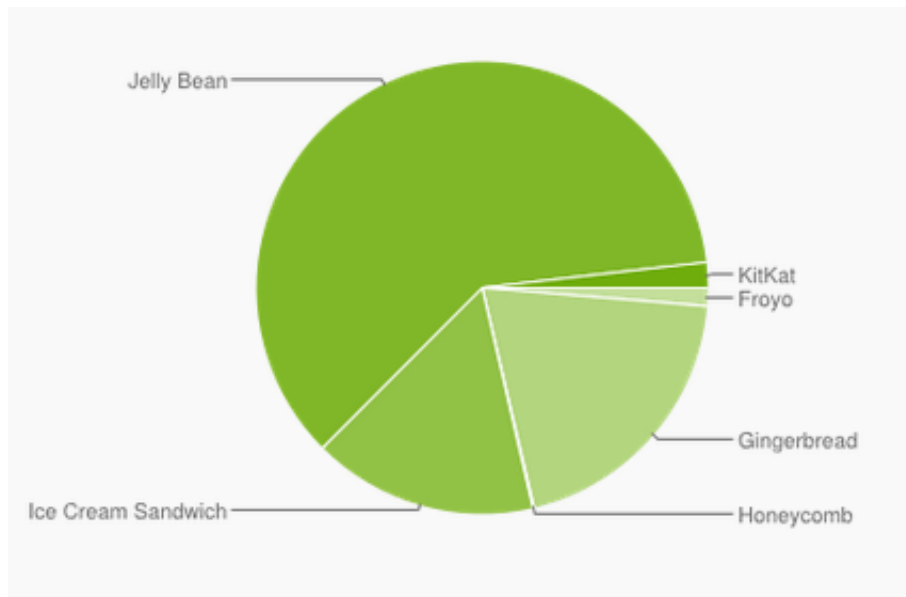


Imagen 3: Distribución de las versiones del SO de Android

Las versiones Apple Pie, Banana Bread, Cupcake, Donut y Ecclair no aparecen en el gráfico ya que fueron las primeras versiones y se consideran obsoletas.

Como podemos ver, el mercado está dominado actualmente por la versión JellyBean. Esta tendencia es posible que cambie debido a que el lanzamiento de KitKat es todavía muy reciente.

Respecto a la cuota de mercado de Android respecto a los otros sistemas operativos para dispositivos móviles, podemos ver, gracias a los siguientes gráficos obtenidos gracias a StatCounter, que actualmente domina el mercado con bastante ventaja.

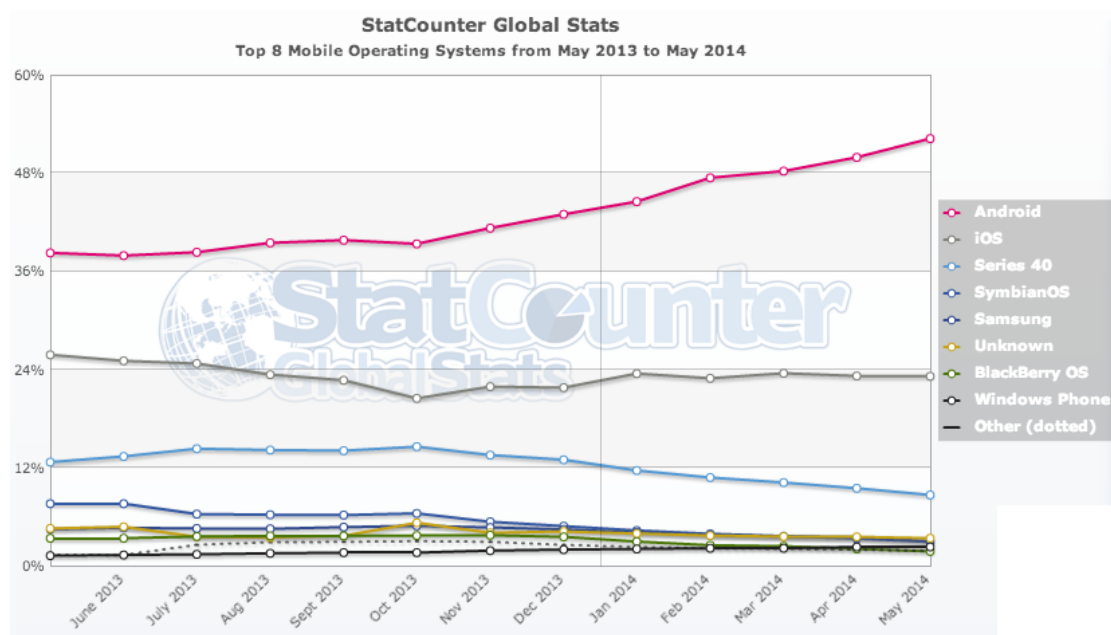


Imagen 5: Estadísticas sobre el uso de SO para móviles en en mundo

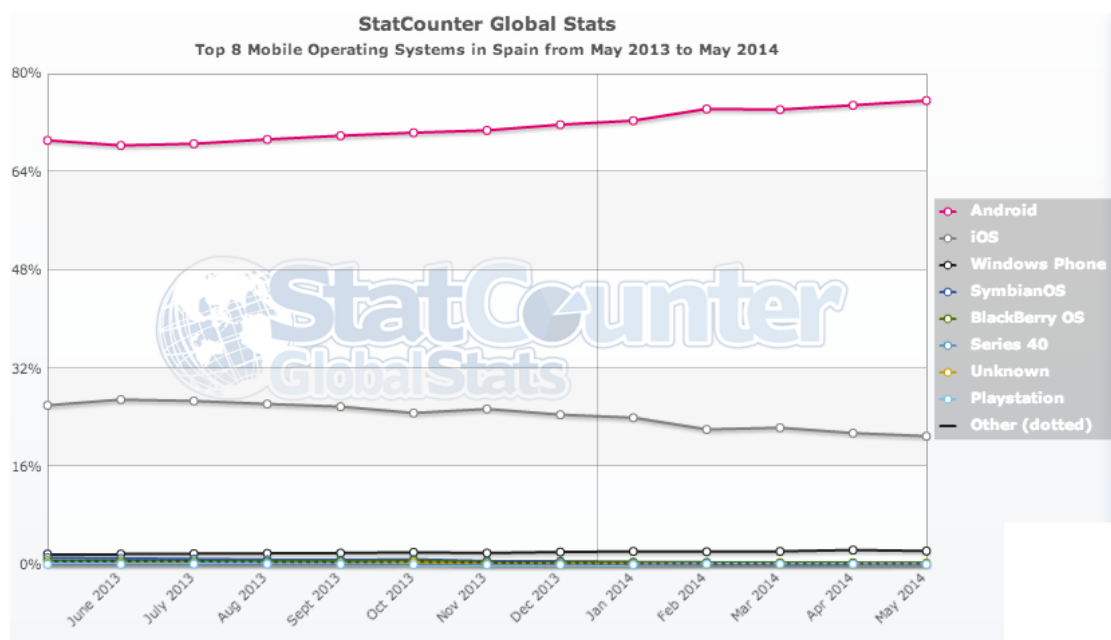


Imagen 5: Estadísticas sobre el uso de SO para móviles en España

En los gráficos, apreciamos que en España, en mayo de 2014, Android domina ampliamente el mercado con un 75.62 % frente al 30% de iOS y la tendencia es que la distancia se haga aún mayor con el tiempo. En cuanto a las estadísticas mundiales,

se puede ver que la tendencia es similar con un 52% para Android, respecto al 23% de iOS.

Cabe destacar que, aunque no lo podamos apreciar en estos gráficos, en USA, mercado muy importante dentro de la telefonía móvil, domina iOS ante Android con un 51% frente al 42% de Android.

2.5 Aplicaciones en Android

Para poder empezar a desarrollar una aplicación en Android, es muy importante conocer, además de la arquitectura que hemos visto anteriormente, una serie de conceptos básicos, o mejor dicho, cuáles son sus componentes

Además de estos conceptos, también veremos que partes componen un proyecto en Android.

2.5.1 Componentes de una aplicación

Procedemos a describir los componentes que podemos encontrar dentro de una aplicación Android:

- *Actividad (Activity)*: Es el componente encargado de ofrecer una pantalla con la que los usuarios de la aplicación pueden interactuar. Debido a ello, tiene siempre una interfaz asociada. Toda aplicación tiene una actividad principal que es la que es lanzada una vez se lanza la aplicación.
- *Ciclo de vida de una actividad*: El ciclo de vida, consiste en los diferentes estados que podemos encontrar en una actividad y los métodos que nos permiten pasar de un estado a otro. En una actividad, podemos encontrar los siguientes estados:
 - *Resumed*: La actividad está en el primer plano para el sistema.
 - *Paused*: La actividad está aún visible, pero hay algún componente que se encuentra por encima.
 - *Stopped*: La actividad no está destruida, pero tampoco está visible.

- *Destroyed*: La actividad está destruida.

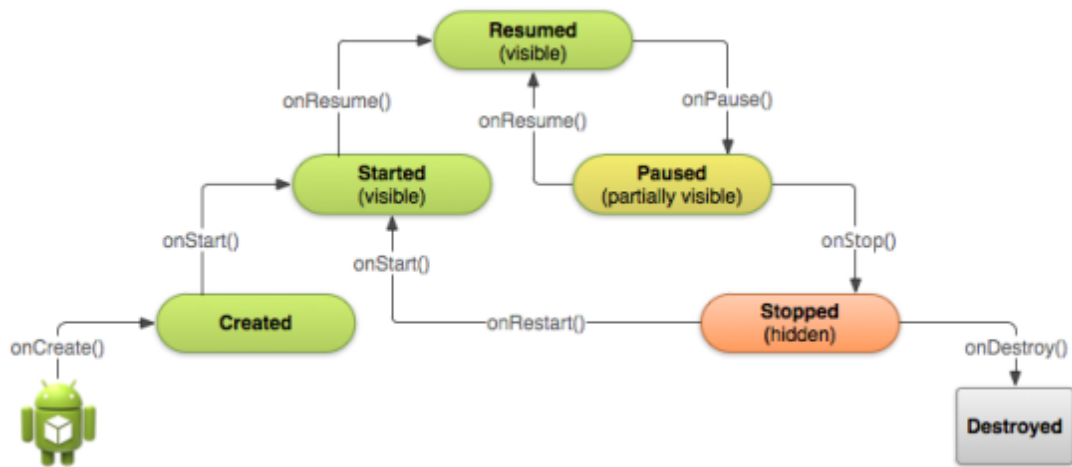


Imagen 6: Posibles estados del ciclo de vida de una actividad en Android

Como podemos apreciar en la figura anterior, para cambiar de estado, la actividad va ejecutando una serie de métodos. En cualquier momento, el programador puede sobrescribirlos para que la aplicación se comporte de la manera deseada. En la siguiente figura, veremos más en detalle que es lo que ocurre en el momento de ejecutar cada uno de estos métodos.

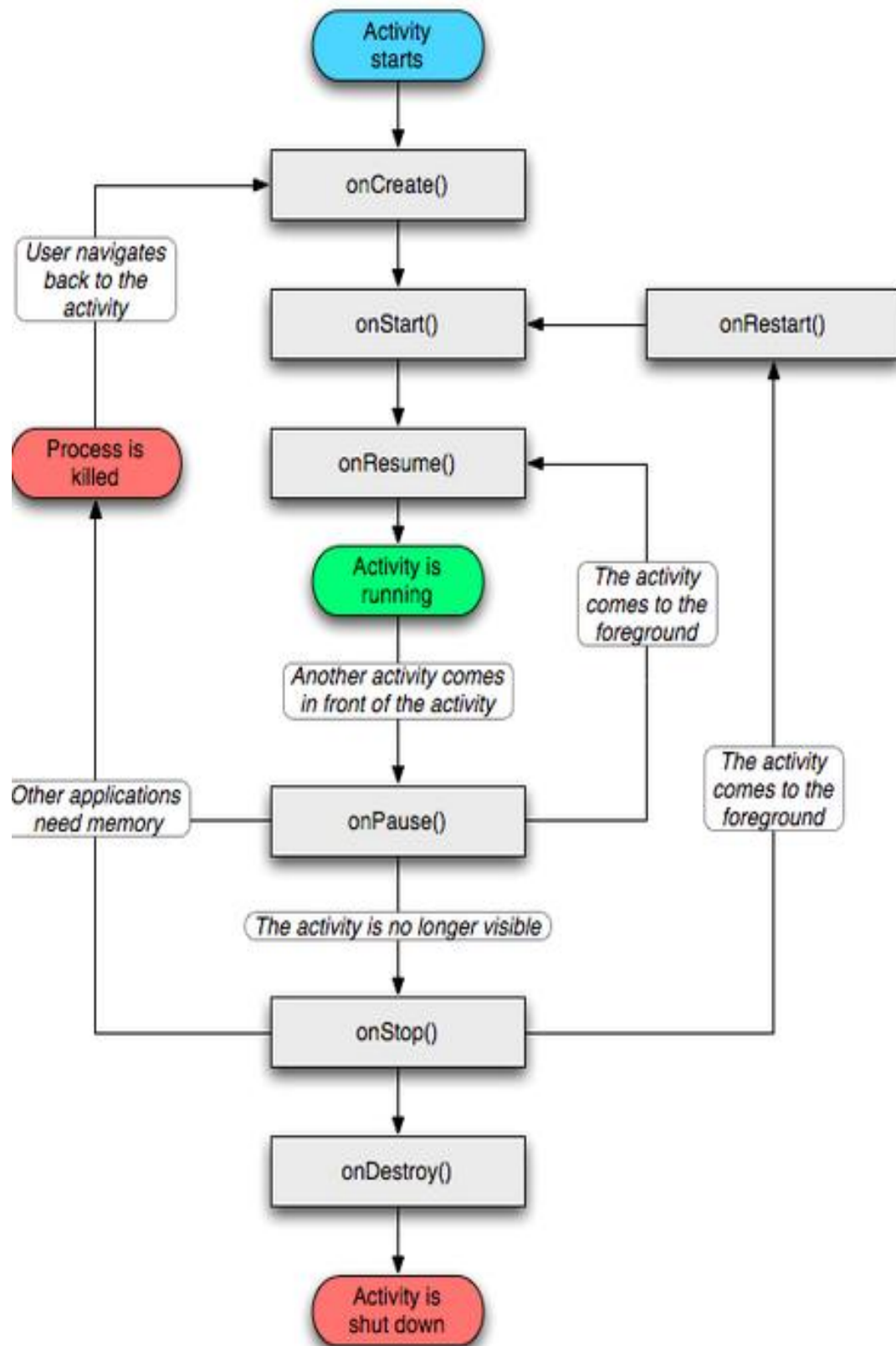


Imagen 7: Ciclo de vida de una actividad en Android

- *View*: Son los componentes que conforman la interfaz gráfica de la aplicación. Android ofrece una gran cantidad de controles como por ejemplo, botones, listas desplegables, imágenes pero que el programador puede modificarlos o crear nuevos desde cero.
- *Intent*: Son las peticiones que son enviadas entre los distintos componentes de la aplicación. Es el elemento básico de comunicación de la aplicación.
- *Service*: Son elementos sin interfaz gráfica que se ejecutan en segundo plano, como por ejemplo la reproducción de sonidos.
- *Contentprovider*: Es el mecanismo que se puede usar para compartir datos entre aplicaciones. De esta manera podemos compartir información sobre nuestra aplicación sin mostrar detalles sobre su estructura o implementación.
- *Broadcastreceiver*: Componente destinado a ejecutar pequeñas tareas en segundo plano destinadas a dar respuesta ante un determinado evento del sistema, como por ejemplo, cuando el dispositivo se está quedando sin batería o cuando un SMS es recibido.

2.5.2 Estructura de un proyecto Android

Una vez creado un proyecto Android vacío desde Eclipse, se generan una serie de archivos y carpetas que serán nuestra base para el desarrollo de la aplicación. Es muy importante no eliminar archivos que creemos que no son importantes ya que posiblemente, nuestra aplicación deje de funcionar. Más adelante, en el capítulo sobre el entorno de trabajo, se hablará más en detalle de la herramienta que ha sido la utilizada para desarrollar el proyecto

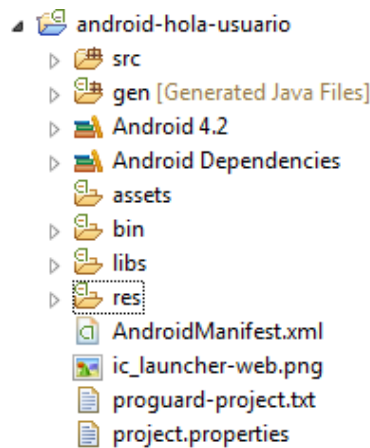


Imagen 8: Estructura de un proyecto Android

- *Carpeta src*: Carpeta que contiene todo el código fuente de la aplicación. Básicamente encontraremos las interfaces gráficas y las clases auxiliares que podemos llegar a usar.
- *Carpeta gen*: Contiene una serie de archivos que se generan y modifican automáticamente. Cada vez que se genera el proyecto, estos archivos se regeneran. Estos archivos están dirigidos al control de los recursos de la aplicación y es muy importante no modificarlos manualmente, ya que podría hacer que nuestra aplicación dejara de funcionar. Dentro de estos archivos que se generan automáticamente, hay que destacar el llamado R.java. Dentro de este archivo, se generan unas constantes que hacen referencia a los recursos contenidos dentro de la carpeta /res/ y su posición de memoria asociada. De esta manera, podremos acceder a esos recursos desde nuestro código fácilmente. Es importante no modificarlo manualmente.
- *Carpeta assets*: Contiene los ficheros auxiliares necesarios para la aplicación como ficheros de configuración o de datos. En nuestro caso es en donde estará situada la base de datos que usaremos en la aplicación. Estos archivos no generan ninguna referencia dentro de la clase R y por lo tanto, la manera de referenciarlos será distinta.
- *Carpeta bin*: Contiene los elementos compilados de la aplicación. Aquí encontraremos el archivo .apk que es el ejecutable que podremos instalar en el dispositivo.

- *Carpeta libs*: Librerías auxiliares que podemos añadir al proyecto en el caso de necesitarlas. Estas librerías están en formato .jar
- *Carpeta res*: Todos los archivos de recursos necesarios para el proyecto, como imágenes, videos, etc. Estos archivos se dividen en las siguientes subcarpetas:
 - *Drawable*: Contiene las imágenes y otros elementos gráficos usados dentro de la aplicación. Se divide en varias subcarpetas dependiendo de la densidad de la pantalla del dispositivo.
 - *Layout*: Ficheros de definición XML de las diferentes pantallas de la interfaz gráfica. Se divide en dos carpetas dependiendo de si el dispositivo está en posición horizontal o vertical.
 - *Anim*: Animaciones usadas dentro de la aplicación.
 - *Color*: Ficheros de definición de los colores usados en la aplicación.
 - *Menu*: Definición XML de los diferentes menús usados en cada una de las pantallas.
 - *XML*: Otros ficheros XML diferentes de los comentados anteriormente.
 - *Raw*: Recursos adicionales usados en la aplicación (por ejemplo, archivos de sonido).
 - *Values*: Ficheros con recursos como el fichero strings, que define las cadenas de texto de la aplicación.
- *Androidmanifest*: Es un archivo situado en la raíz del proyecto. Es básicamente el archivo de configuración de la aplicación. Desde el Androidmanifest podemos definir:
 - *Permisos necesarios*: Permisos que podemos necesitar para interactuar con otras aplicaciones o métodos protegidos (por ejemplo, la vibración del teléfono).

- *Requisitos: Tenemos que declarar el nivel mínimo de la API y las librerías relacionadas que necesitamos*
- *Componentes de la aplicación: Aquí definimos el icono y la etiqueta de la aplicación. También definimos todos los componentes, como actividades y sus acciones relacionadas (intents). Además tendremos que especificar la actividad principal (la primera que se inicia al arrancar la aplicación).*

3. Cocina conmigo y otras aplicaciones

Como ya se ha comentado en la introducción de la memoria, actualmente todo lo relacionado con la gastronomía está al alza. De la misma manera, las aplicaciones para dispositivos móviles relacionadas con la cocina también ha crecido.

Desde este apartado veremos algunos ejemplos de lo que podemos encontrar en el mercado, así como otros productos anteriores que han sentado las bases para otras aplicaciones.

3.1 Cocina conmigo!

El primer ejemplo que encontramos, no se trata de una aplicación para ningún dispositivo, sino un juego para la consola Nintendo DS. Este juego fue lanzado en Japón en 2006 y, debido a su gran acogida y éxito posteriormente se relanzó en Europa, América y Australia en 2008.

La base de Cocina conmigo, es un recetario digital con un total de 245 platos, en donde el usuario es guiado paso a paso a través de un narrador y una voz. Además de un recetario también contaba con características como la toma de notas, filtro de recetas por calorías o un reloj de cocina.



Imagen 9: Imagen del juego ¡Cocina conmigo!

Una parte fundamental de Abre tu nevera! se basa directamente en el funcionamiento de Cocina conmigo, concretamente en el recetario paso a paso, yaque como usuaria del juego lo encontraba muy útil e intuitivo pero contaba con el problema de que no todo el mundo cuenta con una NDS.

3.2 App Canal cocina

Se trata de una aplicación disponible tanto para Android como para Iphone y Ipad creada por la cadena de televisión Canal Cocina.

Es un recetario basado en los programas de su canal. Dependiendo de la plataforma de la aplicación, dispone de unas funcionalidades u otras, siendo la versión para Ipad la que cuenta con las funcionalidades más interesantes.



Imagen 10: Aplicación App canal cocina

En la versión para móvil, sólo podemos buscar y consultar entre las más de 2.000 recetas de que dispone, además de comentarlas y puntuarlas. En cambio en

la versión para *tablet* disponemos de la posibilidad de buscar por listado de ingredientes o gestionar listas de la compra.

3.3 EpicuriousApp

Aplicación que cuenta con más de 8 millones de descargas y está disponible en un número muy grande de plataformas.



Imagen 11: Aplicación Epicurious

Dentro de sus funcionalidades, encontramos el ver receta paso a paso (más de 30.000 platos entre comida y bebida) y gestión de listas de la compra

4. Análisis

4.1 Definición del proyecto

Una vez introducido como funciona Android, además de ver qué otros productos similares podemos encontrar en los diferentes mercados, procederemos a explicar en que consiste **Abre tu nevera!**

Abre tu nevera! es una aplicación desarrollada para Android como hemos comentado anteriormente en otros capítulos que pretende ser funcional y sencilla de utilizar, ya que está pensada para todo tipo de público.

Dentro de la aplicación podremos dividir las funcionalidades en tres grandes grupos: buscar y ver recetas, listas de la compra y mi nevera:

- *Buscar y ver recetas:* se trata de un buscador muy sencillo en el cual podremos buscar entre el listado de recetas disponibles y distinguir entre las que podemos realizar con lo que tenemos en la nevera o en la que necesitamos comprar alguno o todos los ingredientes. Una vez seleccionada la receta que queremos realizar, podremos ver todos sus ingredientes y en el caso de no disponer de todos los ingredientes, podremos crear automáticamente una lista de la compra relacionada con los ingredientes que nos faltan o con todos los de la receta. Una vez vistos los ingredientes, podremos seguir la receta paso a paso. Cada paso consta del texto para su correcta preparación, una imagen que ilustra el paso y, en el caso de tener un tiempo de cocción determinado, un temporizador que puede ser activado en el caso en que el usuario lo precise. Una vez finalizados todos los pasos, se preguntará al usuario si quiere que se elimine alguno de los ingredientes usados de su nevera.
- *Listas de la compra:* es una gestión simple de listas de la compra que el usuario puede crear, modificar y eliminar. Desde aquí, el usuario podrá crear tantas listas como quiera e introducir ítems dentro de ellas. Así mismo, podrá marcar que ítems ya han sido comprados con un simple click e introducir los

ítems de las listas dentro de la nevera para simplificar la gestión. Además podrán visualizarse y modificarse las listas relacionadas con recetas creadas automáticamente.

- *Mi nevera:* Se trata de un pequeño inventario de lo que tenemos en la nevera. Es la base para saber que recetas podemos crear y cuáles no con los alimentos de que disponemos en la nevera. Este inventario, se puede alimentar manualmente introduciendo los ingredientes desde su propio menú o a partir de las listas de la compra creadas también manualmente o creadas a partir de los ingredientes de las recetas.

4.2 Análisis de requisitos

Para poder definir correctamente el proyecto, es muy importante el análisis de requisitos ya que a partir de estos requisitos se efectúa el alcance del proyecto y una mala definición puede hacer que falle la planificación.

Para ello procederemos a describir los requisitos que la aplicación debe cumplir, que al fin y al cabo serán las características principales que definirán el proyecto.

Se han dividido los requisitos en funcionales y no funcionales

4.2.1 Requisitos funcionales

- *Poder buscar una receta entre el listado de recetas.*
- *Poder añadir items y eliminarlos de mi nevera*
- *Poder listar todo lo que hay en mi nevera*
- *Poder crear/eliminar listas de la compra*
- *Poder añadir elemento a las listas de la compra*
- *Añadir elementos de las listas de la compra a la nevera*
- *Filtrar recetas según se tienen los ingredientes o no en la nevera*
- *Generar listas de la compra según una receta y si tengo esos ingredientes o no dentro de mi nevera*

- *Ver receta paso a paso*
- *Poder mover la receta adelante y atrás de manera sencilla*
- *Eliminar ingredientes de mi nevera una vez finalizada una receta*
- *Activar reloj de cocina a partir de un paso*
- *Configurar sonido y vibración del reloj de cocina*

4.2.2 Requisitos no funcionales

Estos requisitos más que definir la funcionalidad de la aplicación, nos definen características que debe cumplir

Interfaz y usabilidad:

La interfaz de la aplicación debe ser intuitiva y fácil de usar para cualquier posible usuario independientemente del nivel técnico que pueda tener. Así mismo, su uso debe corresponderse a los standards de uso de Android.

Diseño:

El diseño de las pantallas debe estar adaptado a la mayor cantidad de dispositivos posibles.

Persistencia:

La información introducida por el usuario debe ser persistente, de manera que pueda seguir consultándola, eliminarla o introducir nueva información.

4.3 Casos de uso

En este apartado podremos ver los diferentes casos de uso de los que consta la aplicación

4.3.1 Diagrama de casos de uso

A continuación podremos ver el diagrama que nos muestra los diferentes casos de uso. Como podemos ver, únicamente tenemos un actor que es cualquier usuario que quiera utilizar **Abre tu nevera!**

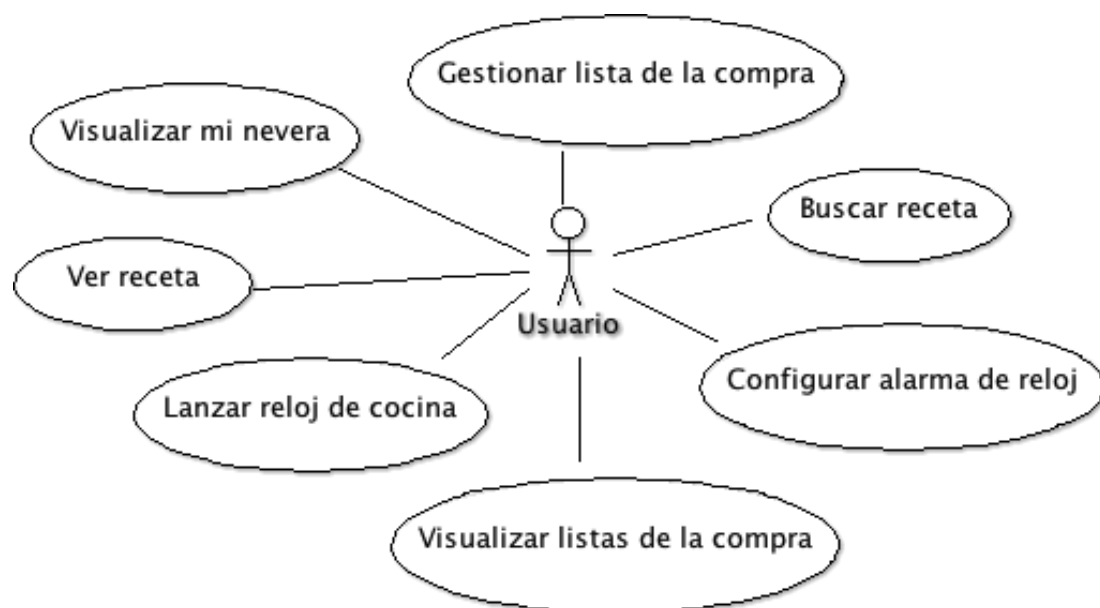


Imagen 12: Diagrama de casos de uso

4.3.2 Casos de uso:

CU1: Buscar receta:

Breve descripción: El usuario, desde el menú principal, selecciona buscar receta y busca de entre las recetas disponibles en la aplicación

Actor principal: Usuario

Flujo principal: Buscar receta

1. El usuario pulsa en el botón *Buscar Receta* desde el menú de la página principal de la aplicación.
2. El sistema muestra el listado de recetas disponibles en la aplicación ordenadas alfabéticamente y un buscador. Por cada una de las recetas, se muestra:
 - 2.1 Una imagen del plato a preparar
 - 2.2 El nombre de la receta
 - 2.3 Un icono que nos dice si está disponible en la nevera totalmente parcialmente, o no está disponible
3. El usuario introduce la receta que quiere buscar en el buscador.
4. El sistema muestra las recetas relacionadas con el texto introducido por el usuario.
5. El usuario pulsa en la receta que quiere visualizar.

Flujo alternativo: No hay recetas relacionadas (lanzado a partir del paso 4)

1. *El sistema no muestra ninguna receta*
2. *El usuario modifica el texto introducido en el buscador*
3. *El flujo continúa en el paso 4*

Flujo alternativo: Cancelar (lanzado a partir del paso 3)

1. *El usuario pulsa el botón de Back*
2. *El sistema vuelve al menú principal.*

CU2: Ver receta.

Breve descripción: Desde el buscador de recetas, el usuario puede iniciar la visualización de la receta paso a paso. El usuario podrá avanzar e ir atrás en cada uno de los pasos.

Pre requisitos: El usuario ha buscado previamente una receta

Actor principal: Usuario

Flujo principal: Ver receta

1. El sistema muestra el paso 0 de la receta:
 - 1.1 Imagen con los ingredientes de la receta
 - 1.2 Listado de ingredientes de la receta
 - 1.3 Disponibilidad de la receta en mi nevera (Disponible, parcialmente disponible o no disponible)
 - 1.4 Icono de crear lista de la compra relacionada
 - 1.5 Botón de siguiente para avanzar en la receta
2. Mientras la receta tenga pasos:
 - 2.1. El usuario pulsa en el botón de *Siguiente*
 - 2.2. El sistema muestra:
 - 2.2.1 Imagen con el paso de la receta
 - 2.2.2 Texto del paso de la receta
 - 2.2.3 Si el paso tiene un tiempo de cocción relacionado, el icono de reloj de cocina
 - 2.2.4 Botones de Siguiente y anterior

3. El sistema muestra un mensaje avisando de que se ha finalizado la receta y un botón para actualizar mi nevera.

Flujo alternativo: El usuario pulsa crear lista de la compra relacionada (lanzado a partir del paso 1)

- 1. El sistema muestra dos opciones para crear la lista, lista completa con todos los ingredientes de la receta o sólo los ingredientes que no tenemos en la nevera*
- 2. El usuario selecciona una de las dos opciones*
- 3. El sistema crea una lista de la compra con los ingredientes de la receta según la opción seleccionada por el usuario.*
- 4. El sistema muestra un mensaje conforme la lista se ha creado correctamente.*
- 5. El flujo continua a partir del paso 2*

Flujo alternativo: Lanzar reloj de cocina (lanzado a partir del paso 2.2.3)

- 1. Caso de uso Activar reloj de cocina.*
- 2. El flujo continua en el paso 2*

Flujo alternativo: Actualizar nevera (lanzado a partir del paso 3)

- 1. El usuario pulsa en actualizar mi nevera.*
- 2. Caso de uso actualizar nevera a partir de receta*

Flujo alternativo: Anterior paso(lanzado a partir del paso 2.2)

- 1. El usuario pulsa un paso en el botón Anterior.*
- 2. El flujo continua en el paso 2 en el paso actual - 1*

CU3: Lanzar reloj de cocina:

Breve descripción: A partir de uno de los pasos de la receta, si este tiene un tiempo asignado de cocción, se puede lanzar el reloj de cocina. El reloj es un temporizador con una cuenta atrás, que, una vez finalizado, avisará al usuario mediante vibración y/o sonido.

Actor principal: Usuario

Flujo principal: Lanzar reloj de cocina

1. El sistema muestra el tiempo y los botones de Iniciar y Parar
2. El usuario pulsa en Iniciar
3. El sistema inicia la cuenta atrás del temporizador
4. Una vez el tiempo se haya terminado, el sistema avisará al usuario mediante vibración y/o sonido de alerta (según la configuración del menú de ajustes)

Flujo alternativo: El usuario pulsa Parar (lanzado a partir del paso 3)

1. *El sistema para el temporizador y lo resetea*
2. *El flujo continua a partir del paso 1*

CU4: Visualizar listas de la compra

Breve descripción: El usuario puede visualizar todas las listas de la compra que tenga introducidas dentro de la aplicación. Estas pueden crearse manualmente o automáticamente a partir de las recetas.

Actor principal: Usuario

Flujo principal: Visualizar listas de la compra

1. El usuario pulsa en el botón Listas de la compra desde la pantalla principal de la aplicación
2. El sistema muestra un listado de todas las listas de la compra almacenadas en el sistema.

CU5: Gestionar lista de la compra

Breve descripción: A partir del listado de listas de la compra, el usuario puede visualizar una lista en concreto y desde ahí gestionar los ítems que la componen .

Pre condición: La lista de la compra seleccionada por el usuario existe.

Actor principal: Usuario

Flujo principal: Gestionar lista de la compra

1. El usuario pulsa en una lista de la compra de entre las que están en el listado.
2. El sistema le muestra un menú con dos opciones, Editar y Eliminar.
3. El usuario pulsa en editar.
4. El sistema muestra, por cada uno de los ítems, un checkbox y el nombre del item.
5. El usuario pulsa en el icono de menú de la pantalla
6. El sistema muestra las siguientes opciones: Añadir item, Marcar todos, Eliminar seleccionados y Añadir seleccionados a mi nevera.
7. El usuario pulsa en añadir item
8. El sistema muestra un input en donde introducir el nombre del ingrediente.

9. El usuario introduce el elemento que quiere agregar a la lista y pulsa añadir.
10. El sistema añade el ingrediente a la lista.

Flujo alternativo: Eliminar lista (lanzando a partir del paso 3)

1. *El usuario pulsa eliminar lista*
2. *El sistema pide confirmación para eliminar*
3. *El usuario pulsa en confirmar*
4. *El sistema elimina la lista y emite un mensaje avisando que la lista se ha eliminado correctamente*

Flujo alternativo: Añadir ítems a mi nevera (lanzando a partir del paso 7)

1. *El usuario selecciona los ingredientes que quiere añadir a mi nevera de la lista y pulsa Añadir seleccionados a mi nevera*
2. *El sistema añade los ingredientes seleccionados a mi nevera y muestra un mensaje avisando que se han añadido correctamente*
3. *El flujo continua a partir del paso 6*

Flujo alternativo: Eliminar ítems (lanzando a partir del paso 7)

1. *El usuario selecciona los ingredientes que quiere eliminar de la lista y selecciona en el menú Eliminar seleccionados*
2. *El sistema elimina los ingredientes y muestra un mensaje avisando que se han eliminado correctamente*
3. *El flujo continua a partir del paso 6*

CU6: Visualizar mi nevera

Breve descripción: El usuario puede visualizar el listado de ítems que tenemos introducidos en nuestra nevera. Estos ingredientes, se pueden configurar manualmente de la manera que describimos o automáticamente a partir de las listas de la compra o finalizando una receta

Actor principal: Usuario

Flujo principal: Visualizar mi nevera

1. El usuario pulsa en el botón Mi nevera desde la pantalla principal de la aplicación
2. El sistema muestra un listado de todos ítems que tenemos introducidos en nuestra nevera
3. El sistema muestra, por cada uno de los ítems, un checkbox y el nombre del ítem.
4. El usuario pulsa en el icono de menú de la pantalla
5. El sistema muestra las siguientes opciones: Añadir ítem, Marcar todos y Eliminar seleccionados.
6. El usuario pulsa en añadir ítem
7. El sistema muestra un input en donde introducir el nombre del ingrediente.
8. El usuario introduce el elemento que quiere agregar a la nevera y pulsa añadir.
9. El sistema añade el ingrediente a la nevera.

Flujo alternativo: Eliminar ítems (lanzando a partir del paso 7)

1. *El usuario selecciona los ingredientes que quiere eliminar de la lista y selecciona en el menú Eliminar seleccionados*
2. *El sistema elimina los ingredientes y muestra un mensaje avisando que se han eliminado correctamente*
3. *El flujo continua a partir del paso 6*

CU7: Configurar alarma de reloj de cocina

Breve descripción: El usuario puede configurar la manera que el dispositivo nos avise de que el tiempo del reloj de cocina ha finalizado

Actor principal: Usuario

Flujo principal: Configurar alarma

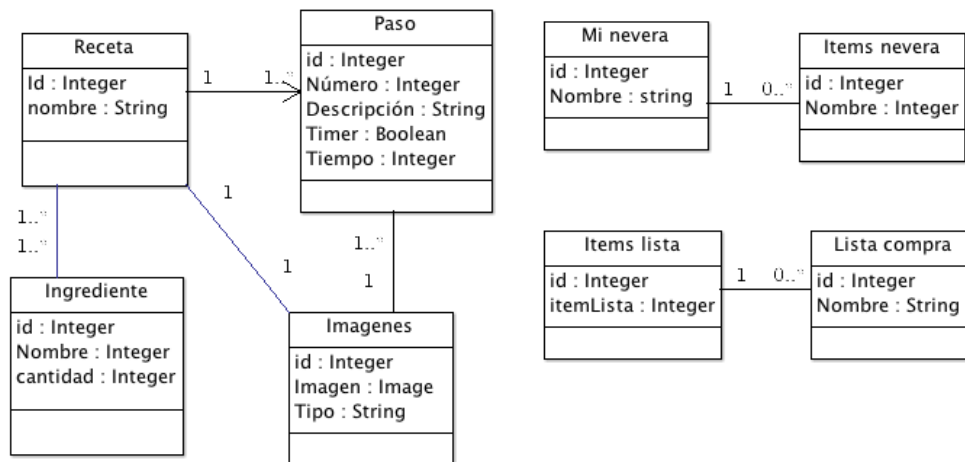
1. El usuario pulsa en el botón opciones desde el menú principal
2. El sistema le muestra las siguientes opciones
 - a. Activar sonido
 - b. Activar vibración
3. El usuario activa / desactiva las opciones deseadas
4. El sistema guarda automáticamente las opciones marcadas por el usuario

4.4 Modelo conceptual

Una vez vistos y teniendo claro que es lo que tiene que hacer la aplicación podemos crear el esquema conceptual.

De esta manera podremos separar fácilmente las clases a implementar, sus atributos y cual es la relación entre ellas.

Este modelo lo mostraremos mediante un diagrama UML.



5. Diseño e implementación de la aplicación.

En este apartado podremos ver como se ha diseñado la aplicación para poder ver como funciona internamente.

Para ello, para empezar veremos como hemos diseñado la base de datos de la aplicación y para entrar un poco más en profundidad, veremos como hemos integrado esta base de datos dentro de la aplicación viendo la clase *Gestor DB* de Abre tu nevera!, que se trata de una extensión de la clase *SQLiteOpenHelper* de Android.

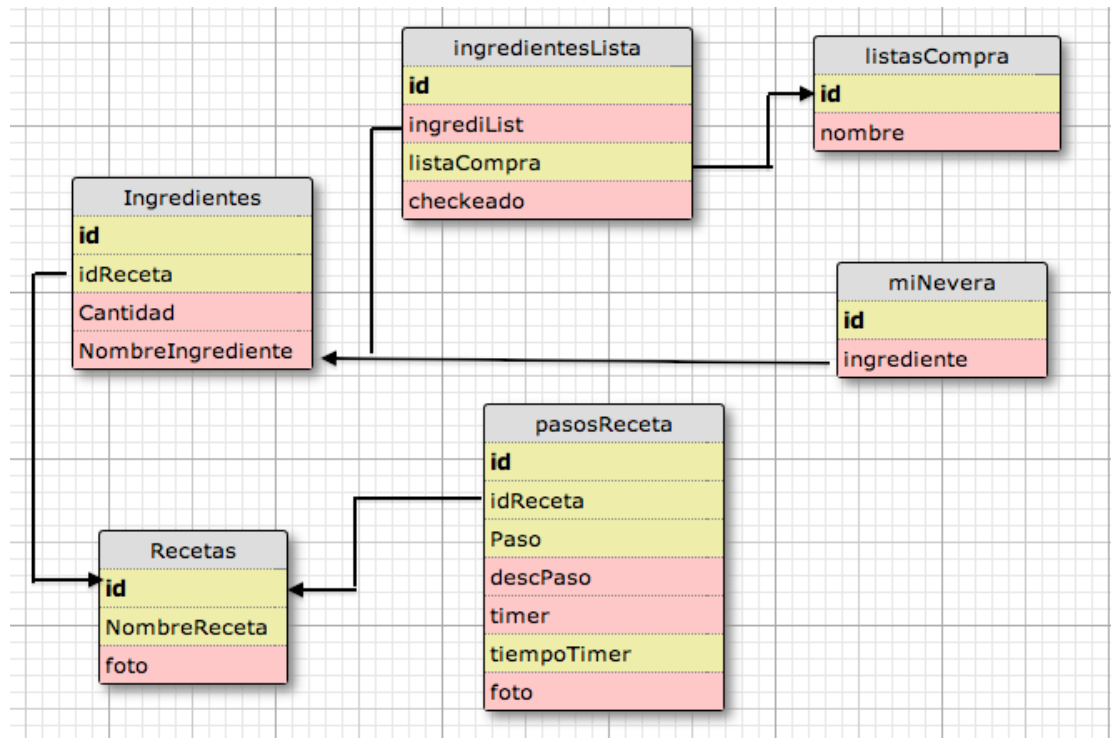
Después veremos los prototipos de la interfaz de usuario que se realizaron al comenzar el proyecto, como se comunican las pantallas entre sí y una vez vistas todas las pantallas finalmente veremos el diagrama de navegabilidad completo. Para finalizar veremos la metodología que he utilizado para crear la aplicación junto con todas las herramientas utilizadas a lo largo del proyecto.

5.1 Diseño de la base de datos y SQLite

Una vez finalizada la parte de la especificación, comienza la parte de diseño.

Una vez analizados los requisitos me di cuenta de que el mayor peso de la aplicación caía tanto en la parte gráfica como en la base de datos relacionada. Por ello vamos a ver el diagrama resultante de crear la base de datos, explicaremos más en detenimiento en que consiste SQLite y la clase que la gestiona dentro de la aplicación.

Como resultado de la especificación, hemos obtenido el siguiente diagrama de base de datos:



En este diagrama se muestran las tablas creadas y la relación entre ellas.

Para acceder a la información y gestionarla, usaremos SQLite.

SQLite, como ya hemos comentado en otros capítulos, es una base de datos que se encuentra integrada dentro de Android. Se trata de una base de datos relacional contenida en una librería bastante pequeña escrita en C.

Su motor no es un proceso independiente con el que se comunique la aplicación sino que su biblioteca se enlaza con el programa de manera que llega a ser parte del mismo.



Imagen 13: Logo de SQLite

Hay dos maneras de gestionar la base de datos SQLite dentro de una aplicación Android: creando las tablas y haciendo los inserts correspondientes para rellenarlas directamente en la clase de Android que gestiona la BD o usar alguna aplicación

complementaria para gestionarla, generar un archivo relacionado que irá en la carpeta assets del proyecto y crear una clase con las que poder usar ese archivo como base de datos en nuestra aplicación.

Debido a que la aplicación de inicio, debe tener un gran volumen de información ya que las recetas deben venir introducidas una vez se instale, descartamos la primera opción, ya que la gestión es muy complicada y es mucho más fácil cometer errores al introducir la información. Por lo tanto, he usado una aplicación auxiliar (SQLite Database Browser, que describiremos más tarde) para ayudar a gestionar la base de datos.

Ya que nuestra base de datos va a venir de un archivo externo, tenemos que crear una clase para que la aplicación sepa que los datos que tiene que mostrar vienen de ese archivo y no intente usar la base de datos que crea automáticamente.

Nuestra clase, debe heredar de SQLiteOpenHelper, que se trata de una clase de Android que nos permite crear la base de datos, actualizar su estructura y sus datos iniciales. Por defecto hay que sobrescribir los métodos onCreate y onUpgrade. onCreate se llama cuando se crea la base de datos por primera vez y es donde se define la estructura de las tablas. OnUpgrade es llamada cuando la versión de la base de datos es distinta a la que había antes y sirve para actualizarla. En nuestro caso, no será necesario rellenar estos métodos ya que toda nuestra información estará en el archivo que está dentro de assets.

Nuestra clase, a la que hemos llamado GestorDB funciona de la siguiente manera:

Primero definimos unas constantes que usaremos dentro de los métodos de la clase:

```
private static final String DB_NAME = "abreTuNeveraDB2";  
private static String DB_PATH = "/data/data/com.example.abretunevera/databases/";
```

La primera, DB_NAME, será el nombre del archivo que se encuentra dentro de assets y que contiene nuestra base de datos. La segunda, DB_PATH, es el path en donde se encuentra la base de datos de la aplicación que siempre se formará a partir de `"/data/data/" + package de la aplicación + "/databases/"`

Una vez definidas estas constantes, tenemos el método de creación de la clase:

```
public GestorDB(Context context) {
    super(context, DB_NAME, null, 1);
    this.myContext = context;
}
```

Al crear la clase, definimos cual es el nombre de nuestra base de datos y le asignamos el contexto.

Una vez definida la clase, nos encontramos con el siguiente método:

```
* Creamos una BD que esté vacía en el sistema y la machaca con nuestro fichero que decimos.
*/
public void createDataBase()
{
    //Si no existe, la copia
    boolean mDataBaseExist = checkDataBase();
    if(!mDataBaseExist)
    {
        this.getReadableDatabase();
        this.close();
        try
        {
            //Copia la bd de la carpeta assets
            copyDataBase();
            Log.e(TAG, "createDatabase database created");
        }
        catch (IOException mIOException)
        {
            throw new Error("ErrorCopyingDataBase");
        }
    }
}
```

Lo que realiza es, comprobar si existe una base de datos con la información de las constantes vistas anteriormente:

```
private boolean checkDataBase()
{
    File dbFile = new File(DB_PATH + DB_NAME);
    return dbFile.exists();
}
```

Y en el caso en que no exista, ejecuta el método copyDataBase():

```
//Copiamos la bd de assets
private void copyDataBase() throws IOException
{
    InputStream mInput = myContext.getAssets().open(DB_NAME);
    String outFileName = DB_PATH + DB_NAME;
    OutputStream mOutput = new FileOutputStream(outFileName);
    byte[] mBuffer = new byte[1024];
    int mLength;
    while ((mLength = mInput.read(mBuffer))>0)
    {
        mOutput.write(mBuffer, 0, mLength);
    }
    mOutput.flush();
    mOutput.close();
    mInput.close();
}
```

Este método busca en assets el archivo definido en la constante DB_NAME y lo copia en el path definido con el mismo nombre que el que se encuentra en la carpeta.

Para usar la base de datos dentro de cualquier activity, deberemos primero, crear una instancia de GestorDB y ejecutar createDataBase. Una vez copiada la base de datos en el caso en que no lo esté, ya estará disponible para usarla en cualquiera de nuestras Activities.

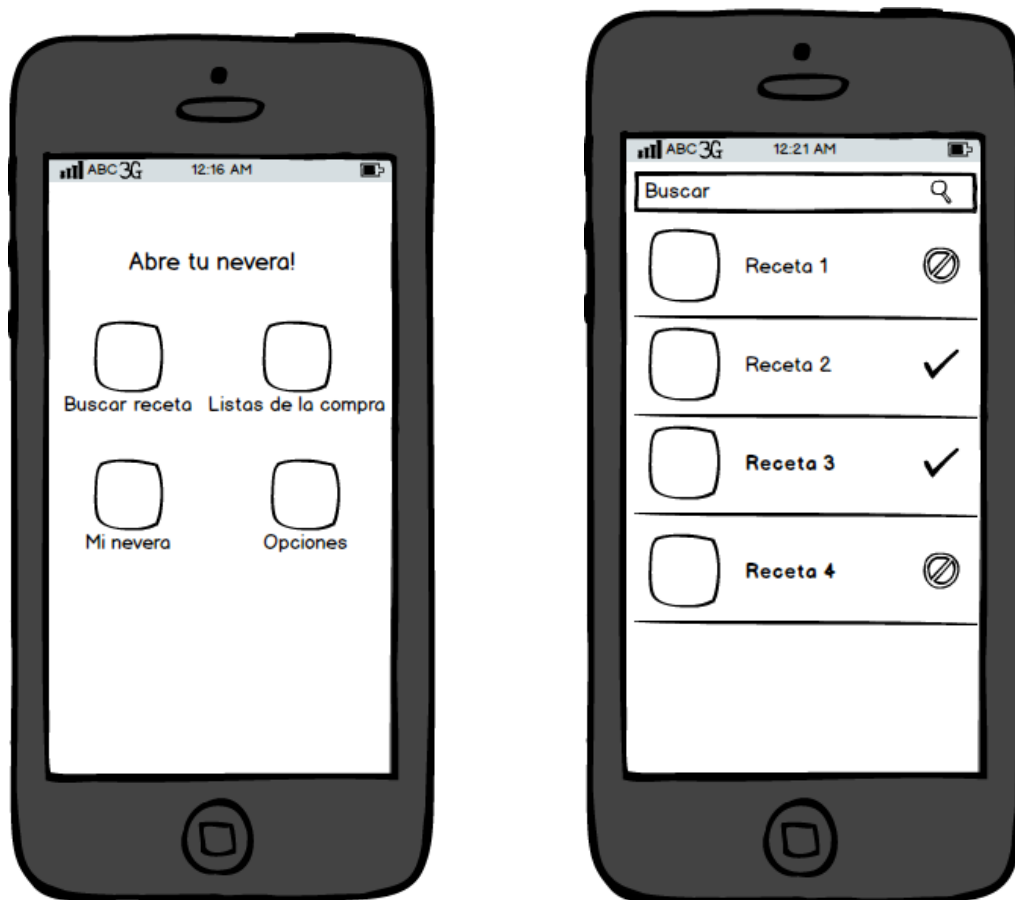
5.2 Prototipos y diagrama de navegación

Antes de proceder al desarrollo de la aplicación, es necesario diseñar las interfaces de usuario mediante prototipos.

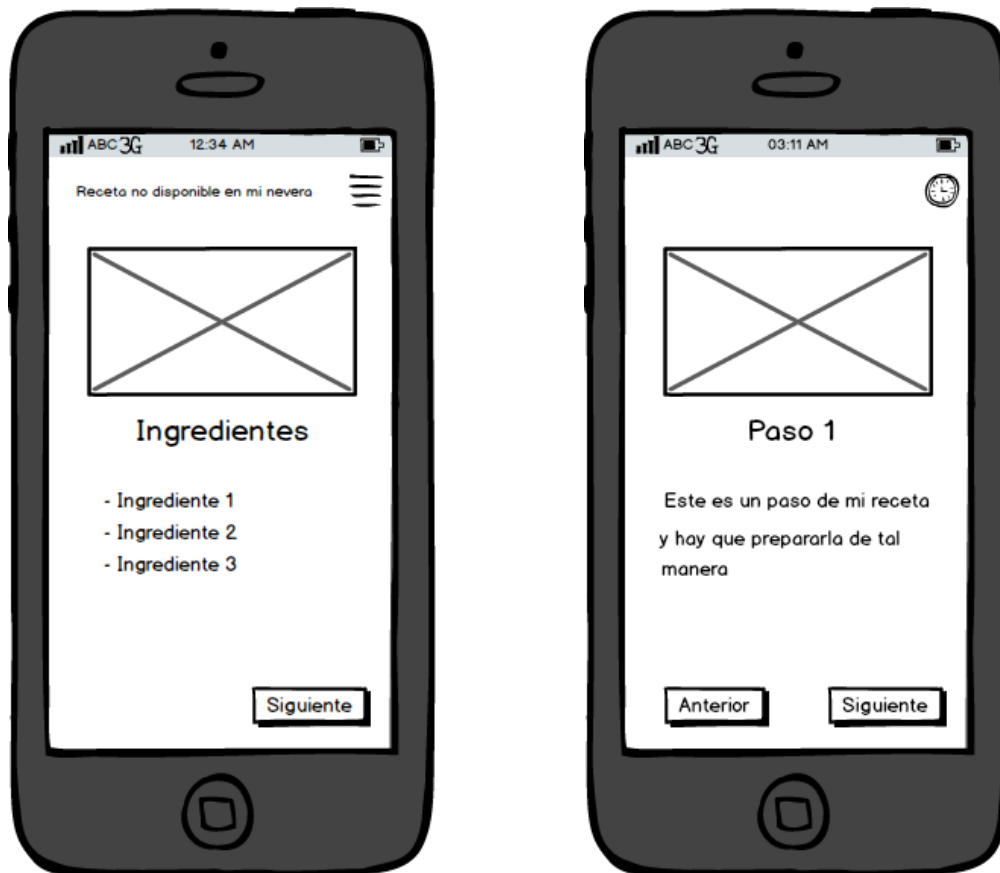
El diseño de la interfaz es uno de los puntos más importantes a la hora de desarrollar cualquier aplicación, pero su importancia se hace más relevante a la hora de ser una aplicación para un dispositivo móvil. Esto es debido a que el tamaño del teléfono es muy limitado y por lo tanto, la distribución de los elementos es más importantes.

La importancia del diseño se debe a que si la aplicación no está bien diseñada y no es usable, el usuario no le prestará atención y terminará por no usarla.

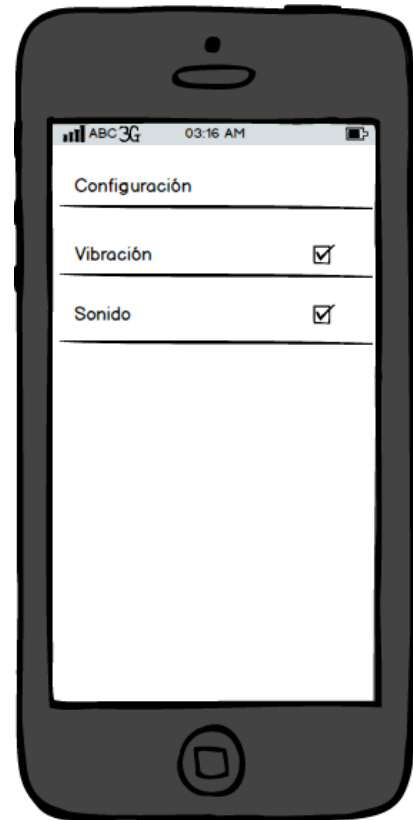
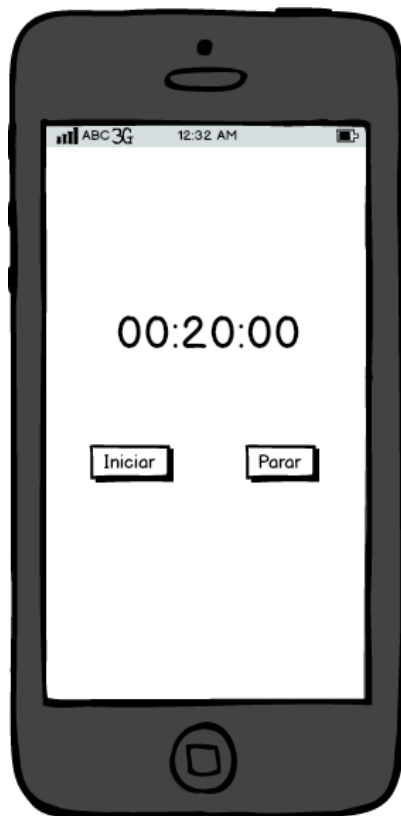
A continuación mostraremos los prototipos de todas las pantallas que conforman Abre tu nevera! y después veremos como se relacionan entre sí mediante un diagrama de navegabilidad:



Aquí podemos ver la página de inicio de la aplicación y la página de buscar receta, en donde podremos ver las recetas disponibles, podremos buscar entre ellas y veremos si están disponibles en mi nevera.



Aquí podemos ver la pantalla de visualización de los ingredientes de una receta, en donde podremos crear listas de la compra relacionadas y la pantalla que se repetirá para cada uno de los pasos de los que conste una receta.



Aquí vemos la pantalla en donde podemos activar el reloj de cocina relacionado con alguno de los pasos de la receta.

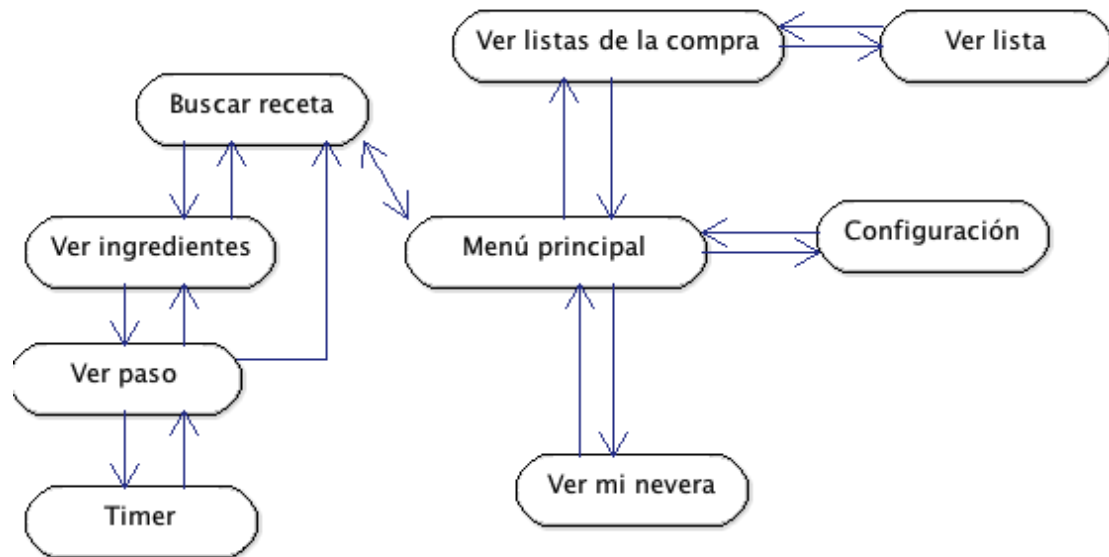
A la derecha, vemos la pantalla de opciones, en donde precisamente, podemos configurar las opciones relacionadas con el reloj de cocina.



Por último, se podemos ver el listado de listas de la compra, una lista de la compra en concreto y el listado de ítems que podemos encontrar en la nevera.

Como se puede apreciar, lo que se ha buscado al diseñar las interfaces ha sido intentar simplificarlo al máximo para que cualquier usuario pueda utilizar la aplicación sin ninguna dificultad.

Una vez vistos todos los prototipos, podemos ver un diagrama de cómo se relacionan todas estas pantallas entre sí:



5.3 Herramientas utilizadas durante el proyecto

A continuación expondré brevemente las herramientas que he usado a la hora de desarrollar todo el proceso. Desde el entorno de desarrollo hasta los diagramas realizados en la especificación.

Eclipse

Eclipse es, actualmente, uno de los entornos de desarrollo más conocidos y más usados por los desarrolladores de software. Se trata de una aplicación multiplataforma y que actualmente cuenta con soporte del kit de desarrollo para Android (Android SDK)



Imagen 14: Logo de Eclipse

SQLite database browser

Se trata de una interfaz gráfica para la gestión de bases de datos SQLite. Desde esta aplicación podremos crear tablas, modificarlas, crear consultas y modificar el contenido de las tablas. De gran utilidad debido a que de por si, SQLite en Android, no cuenta con interfaz gráfica.



Imagen 15: Logo de SQLite Database Browser

ArgoUML

Aplicación para crear diagramas de clase, de secuencia entre otros. Disponible en cualquier plataforma que soporte java.

Balsamiq Mockups

Aplicación usada para crear los prototipos de la aplicación. Esta aplicación es muy sencilla de usar y pueden realizarse prototipos muy completos que pueden semejarse mucho al resultado final. Esta aplicación es de pago, pero para la realización de este proyecto, únicamente se uso la versión de prueba.



Imagen 15: Logo de Balsamiq Mockups

6. Fase de test, depuración de errores y problemas encontrados durante el desarrollo.

Una vez realizada la especificación, el diseño de la aplicación y comenzado la implementación, es importante testear cada una de las funcionalidades que la conforman.

En este capítulo, procederé a describir los resultados de los diferentes tests que me he encontrado tanto al ejecutar la aplicación en los emuladores como en dispositivos reales.

Además de explicar los diferentes problemas que me he encontrado, tanto con Android como con las herramientas que he utilizado para desarrollar la aplicación.

6.1 Pruebas en emulador

Al instalar el SDK de Android dentro de Eclipse, contamos con un emulador propio para ayudarnos en el desarrollo de las aplicaciones.

Este emulador es una herramienta indispensable para cualquier desarrollador, ya que permite simular su ejecución en un entorno seguro además de poder probar varias versiones de Android y varios modelos de dispositivos.

El primer problema que me encontré al empezar el proyecto, fue que el emulador era extremadamente lento ejecutando simplemente el código de ejemplo que se genera al crear un nuevo proyecto Android. Conforme iba avanzando en el proyecto y la aplicación iba creciendo, la velocidad disminuía progresivamente. El hecho de no disponer de una máquina suficientemente potente para la ejecución, agravó este problema de rendimiento.

Asumiendo que el hardware iba a ser el mismo que necesitaba que el emulador funcionara correctamente puesto que era fundamental para el desarrollo del proyecto, decidí buscar una solución e instalar otro emulador que me diera más potencia para poder testear correctamente la aplicación y evitar largas esperas.

6.2 Genymotion y su incorporación a Eclipse

Una vez buscado por la red que posibles opciones tenía para poder instalar un emulador externo, me decanté por Genymotion.



Imagen 16: Logo Genymotion

Genymotion, se trata de uno de los emuladores más rápidos para Android y está basado en el uso de máquinas virtuales x86 optimizadas para ser usadas sobre Virtual box.

Es un emulador que podemos instalar tanto en Windows, Linux como Mac.

Tiene más de 900.000 usuarios registrados usando sus máquinas virtuales para desarrollar aplicaciones para Android. Su uso es muy sencillo e intuitivo y, en el caso de necesitarlo, soporta aceleración de hardware OpenGL.

Genymotion te permite ejecutar imágenes virtuales de Android 4.1, 4.2, 4.3 y 4.4 y cuenta con la posibilidad de poder emular un gran número de dispositivos como Nexus 7, Galaxy Nexus, HTC One, Galaxy Note3 o Moto X.

La incorporación de Genymotion dentro de mi entorno de desarrollo fue bastante sencilla.

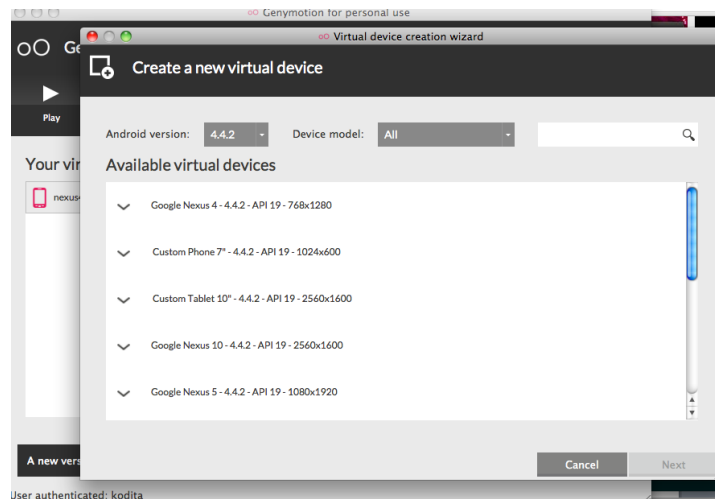


Imagen 17: Creación de un nuevo dispositivo en Genymotion

Una vez descargado Virtual box y Genymotion, tenemos que crear un nuevo dispositivo en Genymotion. Una vez creado, dentro del ADT instalamos el plugin de Genymotion y de esta manera ya podemos tener los dispositivos creados desde Genymotion en nuestro entorno de desarrollo

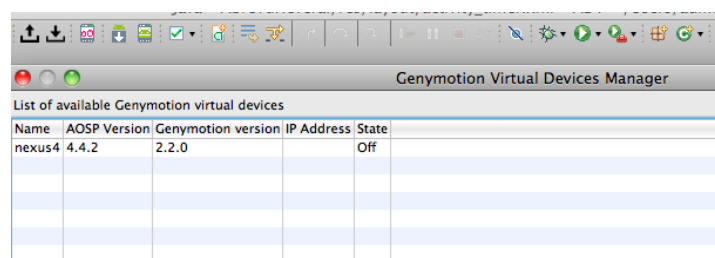


Imagen 18: Virtual devices disponibles en Eclipse

Después de desarrollar la aplicación, puedo decir que el uso del emulador aporta muchas pero también inconvenientes.

Como ventajas, encontramos que el uso del emulador facilita la tarea del desarrollador muchísimo, ya que sería casi inviable el tener que probar cada pequeño cambio en un dispositivo real. Además, necesitaríamos un dispositivo con cada una de las versiones de Android y con diferentes resoluciones para ver que funciona correctamente en cada uno de los diferentes posibles dispositivos. También desde el emulador podemos reproducir sonidos, cosa que personalmente, me ha sido de utilidad para este proyecto.

Como inconvenientes, el emulador, aunque mucho más rápido que el del ADT, sigue siendo más lento que los dispositivos reales. Otro problema que encontramos es

queno todas las funciones son testeables, como por ejemplo la vibración del móvil o el acelerómetro, por lo tanto no se puede probar si la aplicación se ve correctamente o no cuando el móvil se coloca de manera horizontal. Además hasta que la aplicación no se testea en un entorno real, no se aprecian ciertos errores o problemas que procederemos a explicar a continuación.

6.3 Pruebas en móviles reales

Una vez finalizada la primera beta de la aplicación, se procedió a realizar pruebas en móviles reales. Los móviles a los que he tenido acceso, gracias a la colaboración de varios amigos han sido: Samsung Galaxy S, Samsung GalaxyCore y Nexus S. Además de la posibilidad de probar la aplicación en estos dispositivos, también pedí a sus dueños feedback sobre la aplicación para intentar mejorarla y hacerla más usable.



Imagen 18: Dispositivos en los que se ha probado Abre tu nevera!

Todas las pruebas que se han realizado han sido mediante la instalación directa de la apk en el dispositivo.

El primer móvil el que fue testado fue en el Samsung Galaxy S. Se trata del móvil que dispongo actualmente y que mediante unos ajustes previos, tiene instalada la versión 4.4 de Android (Kitkat). Es un móvil que actualmente se ha quedado algo obsoleto ya que sus recursos son bastante limitados en comparación con los dispositivos que podemos encontrar actualmente en el mercado. Realizar las pruebas en este móvil fue positivo, ya que lo más probable era que el rendimiento en otros teléfonos fuera bastante superior.

Una vez instalado en mi teléfono, mandé el archivo a mis testers para que me dieran su opinión. La aplicación parecía funcionó a la primera con buenos resultados, aunque todavía tenía que solucionar algunos primeros bugs.

Los resultados de la primera iteración de pruebas en los dispositivos fueron:

- La carga del listado de recetas era muy lenta en el Galaxy S. En los otros dispositivos resultaba lenta pero soportable.
- La visualización del paso a paso de la receta no funcionaba en el Galaxy S y en los demás dispositivos, funcionaba sólo en algunas ocasiones.
- La vibración del teléfono cuando sonaba la alarma era casi inapreciable.
- La visualización horizontal de la aplicación no era la deseada.
- En cuanto a la usabilidad, los usuarios me dijeron que desde las recetas, era bastante complicado volver al inicio de la aplicación, ya que es posible que un usuario quiera consultar únicamente la lista de ingredientes o ver los pasos y luego volver al inicio sin tener que volver paso a paso hacia atrás. También me comentaron lo que había notado anteriormente, que puesto en horizontal, algunas pantallas se veían fatal. También me comentaron que sería interesante poder ver si la receta estaba disponible en mi nevera sin necesidad de entrar dentro de ella.

Con todo este feedback, me di cuenta de que gran parte del problema era que las imágenes eran demasiado pesadas y que, debido a los tamaños de los móviles, era totalmente innecesario tener fotos tan grandes. Para arreglarlo, redimensioné todas las imágenes y las volví a copiar dentro de mi proyecto.

En cuanto a la visualización horizontal, cree unos layouts especiales para cuando el dispositivo se pusiera en esta posición para poder visualizarla correctamente.

También modifiqué la vibración para que fuera más larga y que el usuario fuera consciente.

Por último, modifiqué los menús de las pantallas para poder volver al índice mucho más fácilmente.

Una vez realizados estos cambios, se realizó una segunda iteración de pruebas:

- La aplicación funcionaba mucho más rápidamente pero habían imágenes que no se veían
- La aplicación seguía viéndose fatal cuando se ponía horizontal.

Con estos últimos resultados, me di cuenta de que había eliminado más imágenes de la cuenta así que tuve que volver a recopilarlas.

Por otro lado tuve que retocar varias veces los layouts hasta conseguir que se viera correctamente. Debido a que el emulador no dispone de vista horizontal, fue un arreglo bastante costoso.

Una vez corregidos estos errores y mejorada la usabilidad según los consejos de mis testadores, di por finalizado el desarrollo de la aplicación.

7. Planificación del proyecto

Este proyecto se inscribió a finales del 2011, pero por motivos tanto personales como profesionales, no lo pude matricular hasta este año.

Por lo tanto el inicio del proyecto comenzó el 24/02/2014.

El proyecto se iba a presentar a finales de Junio, por lo tanto, a partir de esta información, empecé a construir la planificación.

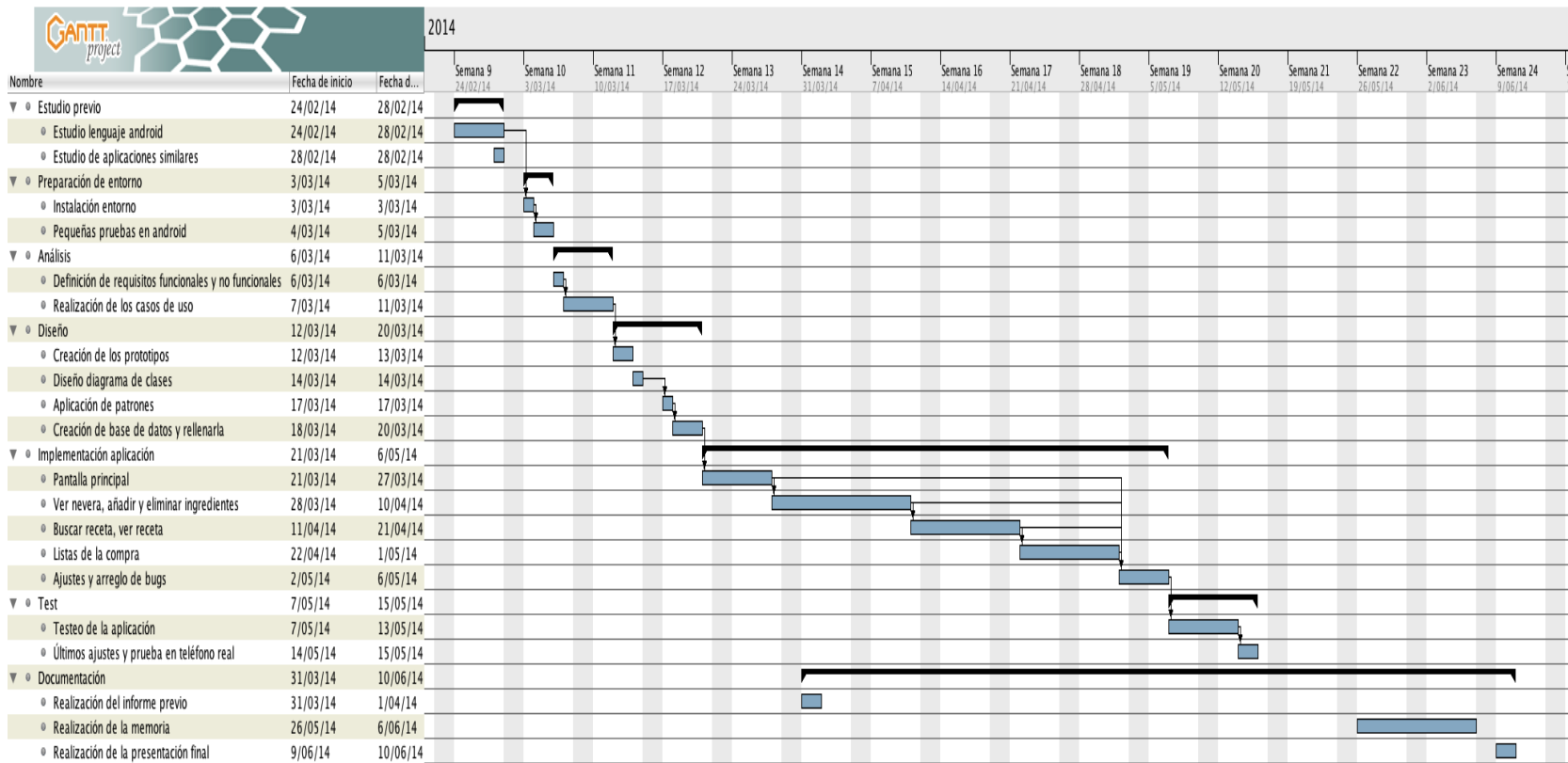
En cuanto a mi experiencia planificando proyectos, ya había realizado planificaciones tanto en la universidad (aunque sinceramente, hace bastante tiempo), como en el entorno laboral. Aunque al ser una temática totalmente distinta y al no tener ninguna experiencia con Android, la planificación inicial fue aproximada. Después veremos que al desarrollar el proyecto mi planificación se desvió bastante respecto a la inicial debido a mi inexperiencia y a los diferentes problemas a los que me tuve que enfrentar.

Lo primero fue calcular el tiempo estimado de realización del proyecto teniendo en cuenta mi disponibilidad y las semanas que había desde el inicio del proyecto hasta el día de entrega.

Contando las semanas, suman un total de 22. Teniendo en cuenta que, trabajando a jornada completa, podía dedicarle 5 horas diarias de lunes a jueves y 6 los viernes, más 8 horas sábado y 5 el domingo. Esta suma hace un total de 39 horas semanales de carga de trabajo pudiendo usar los días festivos en el caso de ser necesarios.

Una vez sabiendo las semanas de trabajo y las horas semanales podía saber que la carga de trabajo del proyecto sería aproximadamente $22 \text{ semanas} \times 39 \text{ horas} = 858 \text{ horas}$

A continuación, podemos ver la planificación inicial del proyecto:



Una vez iniciado el desarrollo, me encontré con el problema del emulador comentado anteriormente en el apartado sobre el test.

Además mi curva de aprendizaje sobre Android fue mayor delo esperado.

Añadido a todo esto, no tuve en cuenta el tiempo que había que invertir a la búsqueda de recetas y su realización, la edición de las fotos y la creación de elementos como los iconos o el fondo de pantalla.

Por lo tanto tuve que reajustar toda mi planificación a partir de mediados de marzo.

Las pruebas en teléfonos reales, también me han llevado más tiempo del que pensaba invertir ya que me he encontrado problemas, que, aunque no muy complicados, requerían bastante tiempo.

A continuación muestro la planificación resultante después de los ajustes que han sido necesarios. La planificación se muestra a partir de mediados de marzo



2014

[illegible]

Como bien podemos apreciar, la diferencia en la planificación es bastante distinta, siendo la primera una planificación en la que tenía bastante tiempo de margen y finalizando en la nueva algo justa de tiempo.

Para finalizar, veremos un desglose de horas según cada uno de los apartados vistos en la planificación:

Grupo de tareas	Horas
Estudio previo	32
Preparación del entorno	24
Análisis	48
Diseño	50
Diseño2 y creación de recetas	90
Implementación	450
Test	80
Documentación	90
TOTAL	864

8. Gestión de costes

Una vez obtenido el número de horas en las que he tardado en desarrollar el proyecto de principio a fin, se puede proceder al cálculo de sus costes.

Debido a que no he usado ninguna herramienta de pago o si lo he hecho, me ha bastado con la versión de prueba, el coste total del proyecto se basa únicamente en las horas de desarrollo.

Para la creación de la aplicación, tampoco he necesitado adquirir ningún elemento hardware externo (no se ha tenido en cuenta el terminal ya que disponía de uno). En el caso en que quisiera publicar la aplicación en la Google Play Store, se tendría que añadir el coste de 25 dolares (aproximadamente unos 18 euros).

Durante la realización del proyecto, se han realizado tareas muy variadas. Normalmente dentro de las empresas, estos diferentes tipos de tareas los realizan personas distintas que cuentan con sueldos distintos. Para intentar acercarme lo más posible a una estimación real, separaremos las tareas por grupo profesional y calcularemos el coste total del proyecto.

Dentro de los diferentes perfiles, podemos encontrar:

- *Analista / diseñador*: Es la persona que se encarga de definir el alcance del proyecto, además de definir también los casos de uso, requisitos funcionales y no funcionales, etc. Una vez finalizado el análisis debe buscar una solución basándose en una tecnología concreta. También es el encargado de escribir la documentación relacionada con el proyecto.
- *Diseñador gráfico*: Es el encargado de realizar los elementos gráficos de la aplicación, como la creación de los iconos o en este caso, la edición de las imágenes.
- *Programador*: Encargado de, a partir de la información y documentación realizada por el analista / diseñador, desarrollar la aplicación. También es el encargado de testear la aplicación.

Consideramos estos perfiles, después de una pequeña búsqueda vemos los precios / hora de cada uno de ellos.

Perfil	Precio / hora
Analista / diseñador	45 €
Diseñador gráfico	30 €
Programador	30 €

Por lo tanto, con esta información, ya podemos calcular el coste total del proyecto

Grupo de tareas	Precio hora	Número de horas	Precio total
Estudio previo	30 €	32	960
Preparación del entorno	30 €	24	720
Análisis	45 €	48	2.160
Diseño	45 €	50	2.250
Diseño2 y creación de recetas	30 €	90	2.700
Implementación	30 €	450	13.500
Test	30 €	80	2.400
Documentación	45 €	90	4.050
TOTAL			28.740

Como podemos apreciar, este precio no acaba de ser del todo realista, ya que en una empresa real en donde se puede realizar un proyecto de esta envergadura, habría profesionales que dominarían la materia mucho más que yo y además, las tareas de investigación y el tiempo invertido en la resolución de problemas con la plataforma no existiría, por lo tanto el presupuesto se vería reducido a aproximadamente la mitad, lo que se correspondería mucho más con la realidad. Por otro lado,

se deberían tener en cuenta en este segundo caso, el de una empresa que desarrolle la app, los costes derivados de su propia estructura.

9. Posibles ampliaciones

Al definir el alcance del proyecto, se tiene que tener en cuenta que el tiempo de realización del proyecto es muy limitado, por lo que al ser una aplicación creada desde cero, siempre quedan ideas que, por tiempo, al final no hay más remedio que descartarlas.

A continuación veremos varias ideas que se contemplaron en un inicio y que, nunca se sabe, podrían estar en una segunda versión de la aplicación:

- Ampliar el número de recetas: Debido a que el tiempo de realización del proyecto está muy acotado, el número de recetas disponibles en la primera versión de Abre tu nevera! es muy limitado ya que el coste de añadir una receta es bastante costoso (hay que investigar para crear las recetas, crear el paso a paso, realizar las fotos y ajustarlas para que se vea correctamente). Lo ideal sería tener aproximadamente unas 100 recetas para que el usuario tuviera un amplio abanico de opciones para realizar recetas.
- Búsqueda avanzada (añadir varios filtros de búsqueda en la búsqueda de recetas): una vez cumplida la primera ampliación, sería interesante añadir filtros como tipo de plato (primero, segundo, postre o bebida), dificultad, carga calórica o si es apto para vegetarianos / veganos, celíacos o intolerantes a la lactosa.
- Añadir productos a mi nevera a partir de códigos de barras: La idea sería integrar dentro de la aplicación un lector de códigos de barras que a partir de un producto, pudiera añadirse automáticamente a mi nevera. Esta opción también estaría disponible dentro de las listas de la compra, en donde podría relacionarse un ítem de la lista con un código de barras y de esa manera, al añadirse a la nevera, podríamos tener el producto en concreto.
- Mis recetas: La idea es que el usuario pueda crear sus propias recetas y de esta manera integrarlas con los ingredientes de Mi nevera. Así el usuario podrá ver si tiene los productos necesarios para poder realizar las recetas que cocina día a día.

- Integración con redes sociales: Una vez que el usuario pueda crear sus propias recetas, sería interesante que fuera posible compartirlas con otros usuarios mediante redes sociales como Facebook. Además que otros usuarios pudieran integrar recetas de otros usuarios a su aplicación.
- Mis favoritos: Esta opción podría estar disponible tanto en la búsqueda de recetas como en las listas de la compra.

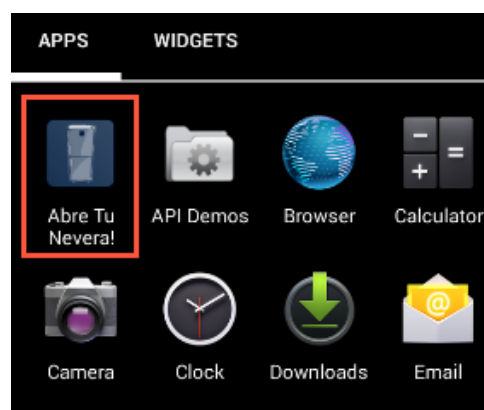
En cuanto a las recetas favoritas, sería una manera de tener acceso de una manera más sencilla a las recetas que más nos gustan o las que solemos hacer más a menudo.

En cuanto a las listas de la compra, es común que al realizar la compra, más o menos se compren una lista similar de productos cada semana. De esta manera, no tendríamos que crear una lista para cada vez sino que la tendríamos siempre a mano.

10. Manual de usuario

Una vez visto todo el desarrollo del proyecto, veremos el manual de usuario de la versión final de Abre tu nevera! que, a parte de una guía para toda persona que quiera usar la aplicación, pretende mostrar el resultado final de todo el proceso de desarrollo.

Una vez instalada la aplicación, veremos que se nos ha creado el icono dentro de nuestro dispositivo:



Al pulsar en el icono, veremos la pantalla de inicio de la aplicación:



Podemos ver 4 apartados diferentes, *Buscar receta*, *Listas de la compra*, *Mi nevera* y *Opciones*.

Mi nevera

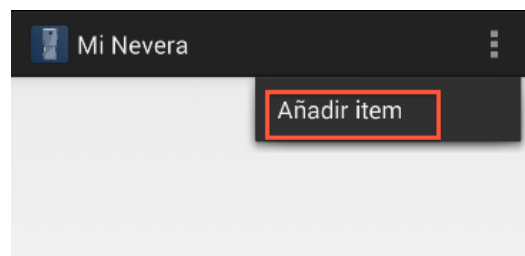
Mi nevera pretende ser un inventario de los productos que tenemos en casa. A partir de la información que se encuentra en mi nevera, podremos ver que recetas tenemos disponibles para cocinar.

La consulta de Mi nevera también es muy útil a la hora de realizar la compra ya que podremos comprobar al momento que productos tenemos disponibles y cuales no en casa. Así evitaremos comprar productos con los que ya contamos (problema que seguro que le ha ocurrido a todo el mundo).

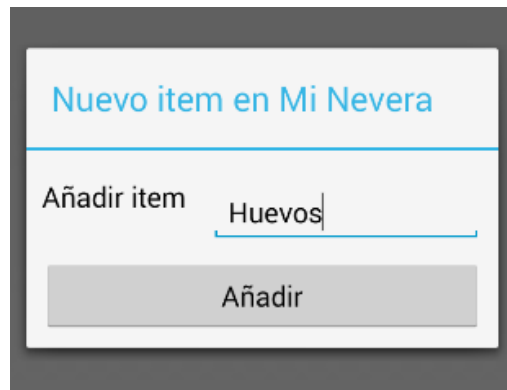
Para acceder a Mi nevera, sólo tenemos que pulsar el icono en la pantalla principal.

Al acceder por primera vez una vez instalada, veremos que no encontramos ningún elemento (tenemos la nevera vacía).

Para añadir un nuevo elemento a la nevera pulsaremos en el icono de menú y seguidamente, pulsaremos en Añadir Ítem:



Una vez pulsado el menú, nos aparecerá una pequeña ventana por encima en donde deberemos indicar el nombre del producto que tenemos:



Una vez escrito el item que queremos introducir en Mi Nevera, pulsaremos añadir y automáticamente aparecerá el producto dentro de la lista:



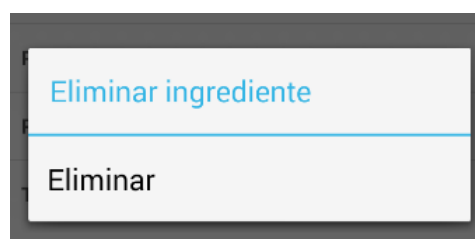
Los productos introducidos, se ordenan automáticamente por orden alfabético para facilitar el visionado al usuario.

Podemos realizar esta operación tantas veces como sea necesario hasta obtener un listado completo de todo lo que tenemos.

Esta lista se puede actualizar automáticamente a partir de las listas de la compra.

Veremos esta funcionalidad más adelante.

En el caso en que queramos eliminar un elemento de la lista porque nos hemos equivocado al introducirlo o simplemente, se nos haya terminado y no disponemos de el, sólo tenemos que pulsar en el elemento a eliminar:



Una vez confirmada la eliminación, la lista se actualizará automáticamente sin el elemento que acabamos de quitar de la lista

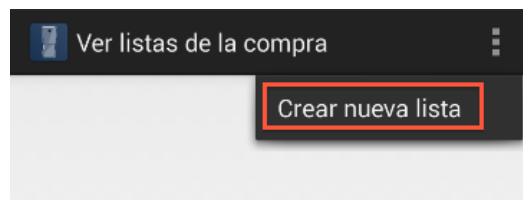
Listas de la compra

En este apartado de la aplicación, podremos crear y gestionar listas de la compra para que el usuario pueda organizarse correctamente a la hora de ir a comprar. Además veremos como poder añadir elementos a Mi Nevera a partir de listas de la compra creadas previamente.

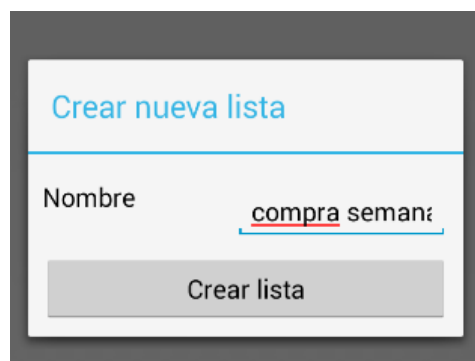
Para acceder al apartado de Listas de la compra, sólo tendremos que pulsar en su icono.

Una vez dentro, como en el caso de Mi nevera, veremos que no encontramos ningún elemento (ya que no tenemos creada ninguna lista de la compra por defecto).

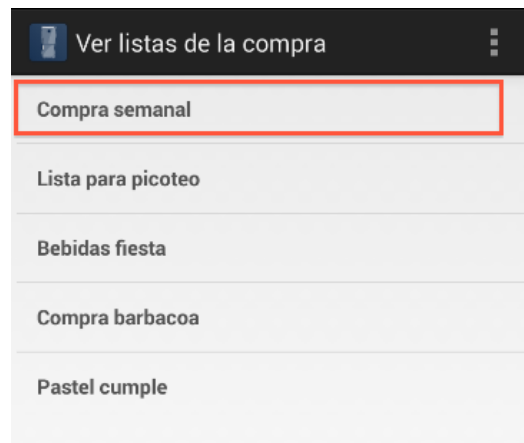
Para crear una lista, el método es muy similar al de añadir un item dentro de Mi nevera y deberemos pulsar en el menú y a continuación crear nueva lista:



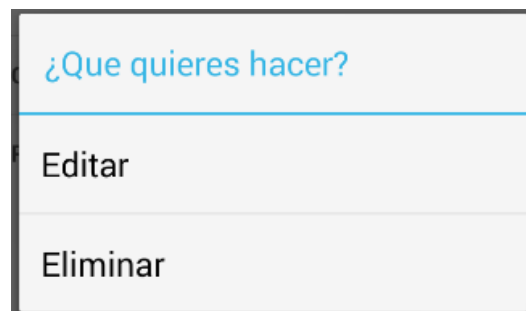
Una vez pulsado, nos aparecerá una pequeña pantalla en donde deberemos añadir el nombre de la lista que queremos crear:



Para terminar, pulsaremos en crear lista. Una vez pulsemos, automáticamente aparecerá nuestra nueva lista dentro del listado de listas de la compra. En este caso, las listas aparecerán ordenadas según cuando se hayan creado, siendo la creada más recientemente la que esté primero:



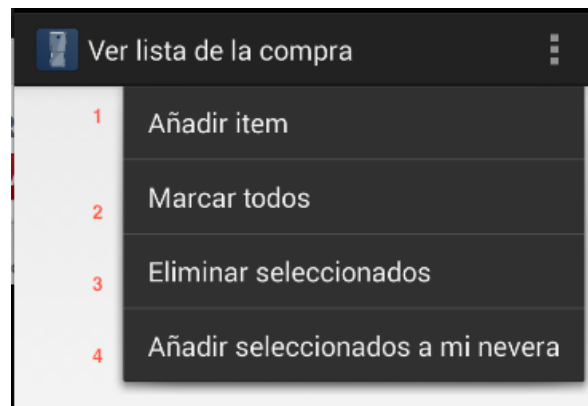
Una vez creada, para realizar cualquier acción relacionada sobre una lista, pulsaremos en la línea en donde está situada. Una vez pulsada, nos aparecerá el siguiente menú:



Pulsaremos en editar para poder gestionar los elementos de la lista o eliminar si ya no necesitamos la lista que hemos creado anteriormente.

Una vez pulsado editar, si es una lista que acabamos de crear, vemos que no tenemos ningún elemento relacionado.

Pulsaremos en el menú de la pantalla para ver todas las funcionalidades que tenemos relacionadas:



1. *Añadir item*: Para añadir elementos a la lista de la compra, pulsaremos en añadir item. Una vez pulsado nos aparecerá una pequeña pantalla en donde podremos añadir el elemento que deseamos añadir a la lista de la compra:



Una vez pulsado añadir, se añadirá el item a la lista de la compra. Podemos repetir esta operación tantas veces como elementos necesitemos en la lista.



Como vemos en la figura, cada elemento tiene un checkbox asociado. Esto nos sirve para indicar que productos ya hemos adquirido y cuales no. Cada vez que marquemos un elemento, se guardará, de manera que aunque salgamos de la lista y volvamos a entrar, los elementos que hemos marcado anteriormente, seguirán marcados.

Esta operación también se puede automatizar a partir de las recetas paso a paso. Esta funcionalidad la veremos más adelante.

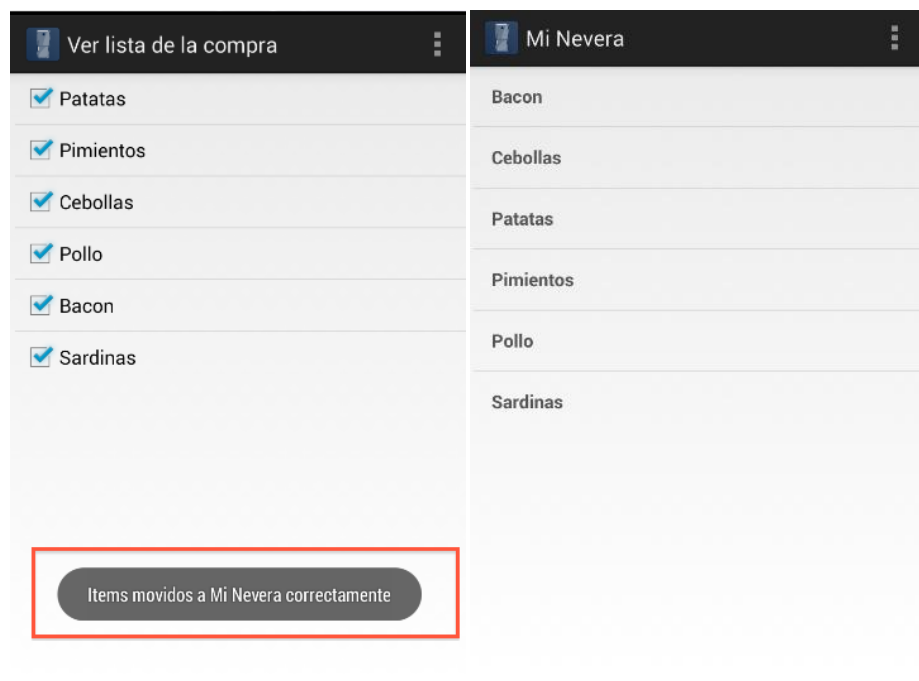
2. *Marcar todos:* Es posible que vayamos a la compra y en ese momento no queramos sacar el móvil cada vez que añadimos un producto al carro. De manera que, si hemos comprado todo lo que teníamos en la lista, para optimizar el proceso de marcar todos, podemos pulsar en este ítem del menú. Una vez pulsado, podremos ver que todos los elementos que tenemos aparecen señalados.



3. *Eliminar seleccionados*: Si queremos eliminar elementos que ya hemos comprado y no los queremos tener en la lista o simplemente nos hemos equivocado al añadir uno de los elementos, podemos marcar los que queremos eliminar y pulsar este punto de menú.
Automáticamente, los elementos que hemos marcado desaparecerán de la lista de la compra.
4. *Añadir seleccionados a mi nevera*: si hemos adquirido alguno de los productos que hemos indicado dentro de la lista, podemos añadirlos automáticamente a mi nevera con un sólo click.

Una vez marcados los productos (puede ser únicamente uno o todos los de la lista), pulsaremos en Añadir seleccionados a mi nevera.

Una vez pulsado, la aplicación nos mostrará un mensaje conforme los productos han sido añadidos a Mi nevera:



Como podemos ver, los elementos marcados previamente desde la lista de la compra, se han añadido correctamente a mi nevera.

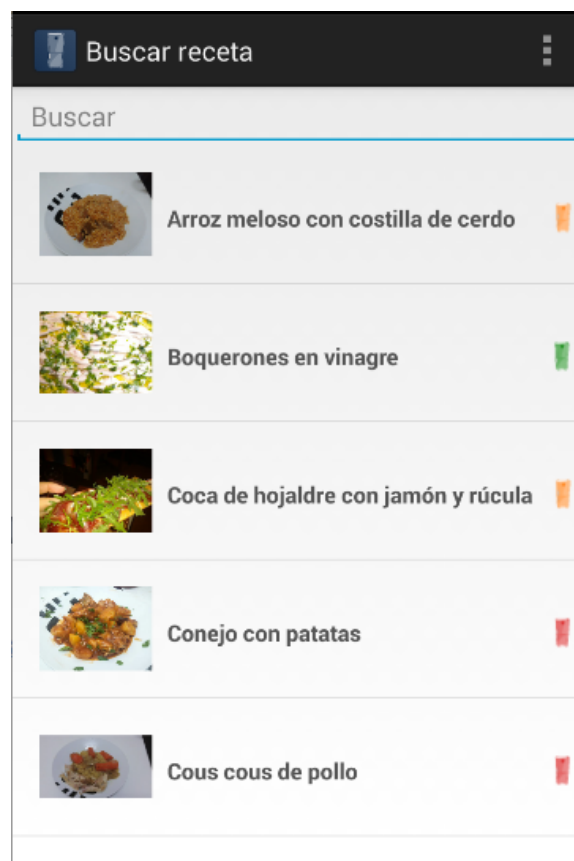
Buscar receta

Desde esta parte de la aplicación, podremos ver las recetas disponibles de Abre tu nevera! y las podremos seguir paso a paso. Además de un vistazo podremos ver

querecetas podemos realizar con los productos que tenemos en la nevera y en el caso de no disponer de ellas, podremos hacer automáticamente una lista de la compra con los ingredientes que nos falta. Una vez terminada la receta, en el caso de haberla realizado, podremos actualizar nuestra nevera con los productos que se nos han terminado.

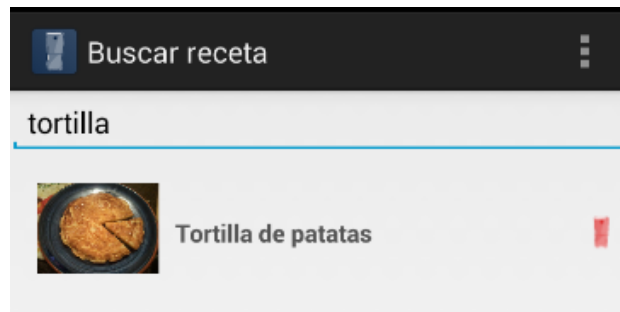
Para acceder a este apartado, pulsaremos el botón Buscar receta desde el menú principal.

Una vez dentro podremos ver la siguiente pantalla:



Aquí podremos ver el listado de recetas disponibles en Abre tu nevera!

Disponemos de un buscador para poder facilitar la búsqueda de la receta deseada:



Por cada una de las recetas, encontramos 3 elementos:

- Foto descriptiva del resultado final de la receta
- Nombre de la receta
- Icono de disponibilidad en Mi Nevera. Si este icono es rojo, significa que no disponemos de ningún ingrediente en mi nevera, si es naranja significa que disponemos de algún ingrediente y si es verde significa que podemos realizar la receta con lo que encontramos en Mi nevera.



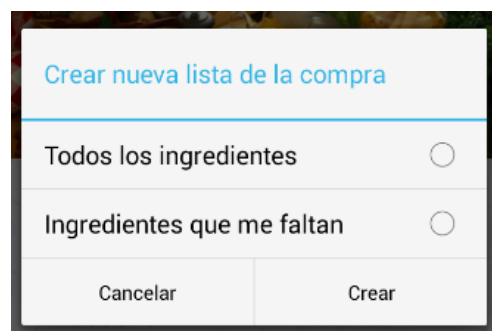
Una vez elegida la receta que queremos realizar, pulsamos en la línea que la representa.

Una vez pulsada, aparecerá la siguiente pantalla:

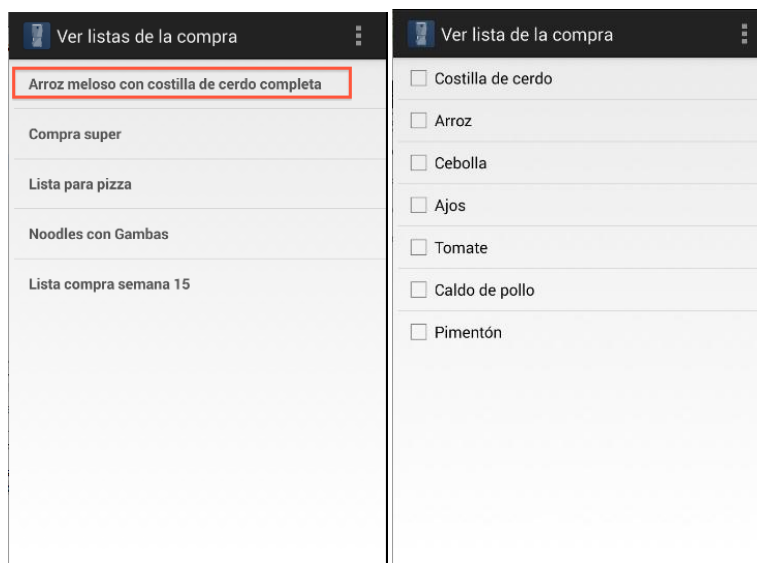


Esta pantalla encontramos:

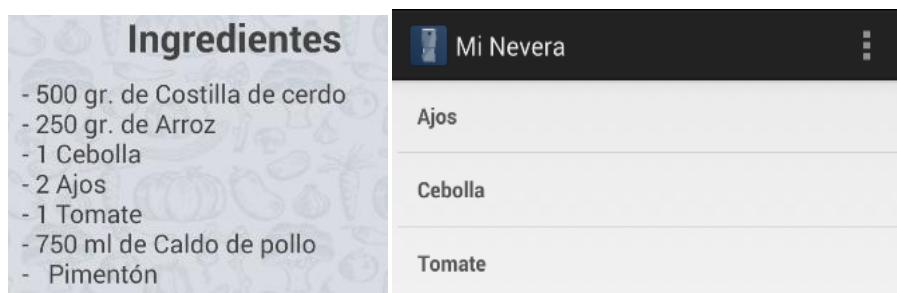
1. *Ingredientes*: Encontramos el listado de ingredientes que vamos a necesitar para realizar la receta. Los ingredientes son los orientativos para realizarla para unas 4 personas.
2. *Disponibilidad de la receta y crear listas de la compra asociadas*: En la parte superior de la pantalla, nos podemos encontrar tres mensajes, *Disponible en Mi nevera*, *Parcialmente disponible en Mi nevera* o *No disponible en mi nevera*. En el caso de encontrar cualquiera de los dos últimos, encontraremos el icono de una lista en la esquina superior izquierda. Este icono sirve para crear una lista de la compra relacionada con los ingredientes de la receta. Si pulsamos en el icono, encontraremos el siguiente menú:

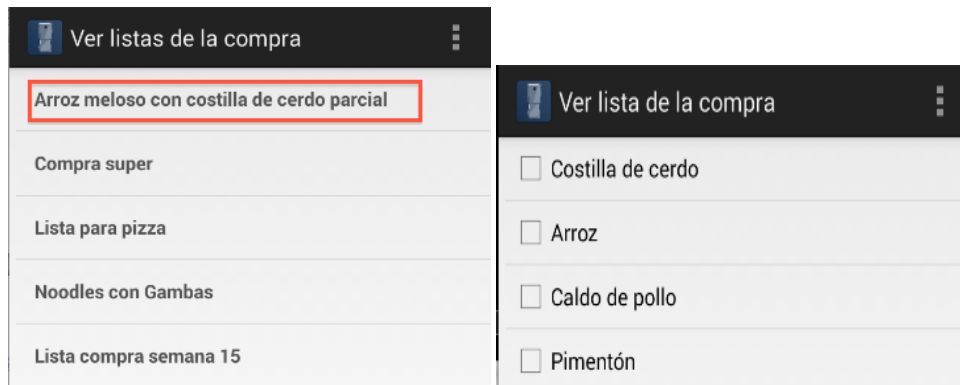


Si seleccionamos la primera opción y pulsamos crear, se nos creará en el apartado *Listas de la compra* una lista con el nombre de la receta mas la etiqueta completa, es decir, si creamos una lista para la receta “Arroz meloso con costilla de cerdo”, el nombre de la lista sería “Arroz meloso con costilla de cerdo completa”. Una vez creada, la aplicación nos mostrará un mensaje conforme la lista se ha creado correctamente. Esta lista contendrá todos los ingredientes listados en la pantalla y que son necesarios para realizar la receta:



En el caso de seleccionar la segunda opción (ingredientes que me faltan), la aplicación hace un matching entre lo que tenemos en la nevera y los ingredientes de la receta y nos crea una lista de la compra con el nombre de la receta + “parcial” que contiene todos los ingredientes de la receta que no ha encontrado en Mi nevera:





3. *Botón siguiente*: Para iniciar el paso a paso de la receta, deberemos pulsar en el botón siguiente. Una vez pulsado, la receta empezará en el paso 1.

Una vez iniciado el paso 1, podremos visualizar la siguiente pantalla:



1. *Foto del paso*: Cada uno de los pasos tiene una imagen que ilustra lo que tiene que hacer el usuario para seguir la receta

2. *Número de paso:* Para que el usuario sepa en que punto de la receta está, se numeran los pasos. Todas las recetas empiezan por el paso 1
3. *Texto del paso:* Descripción del paso para seguir la receta. El texto se complementa con la imagen que se muestra más arriba
4. *Botones de anterior y siguiente:* El usuario en cada momento puede avanzar y retroceder durante el proceso de preparación de la receta. En el caso de estar en el paso 1, al pulsar en anterior, se volverá a la lista de ingredientes.
5. *Reloj de cocina:* En el caso que la receta tenga un tiempo asignado (en este caso, 1 hora) aparecerá el icono del reloj. Al pulsarlo, tendremos disponible un reloj con una cuenta atrás con el tiempo asignado al paso de la receta:



Mediante los botones de iniciar y parar podemos controlar el tiempo.

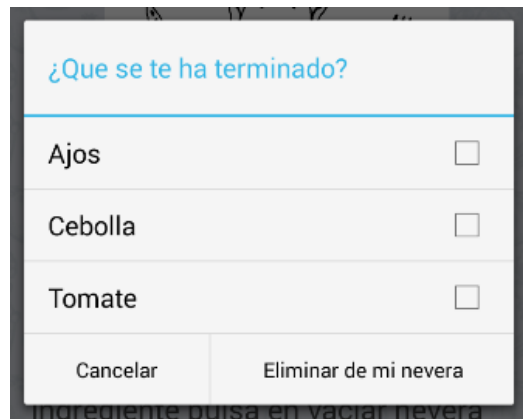
Una vez pasado el tiempo del paso, la aplicación nos avisará mediante el sonido de una alarma y la vibración del móvil.

Una vez visualizados todos los pasos de la receta, llegaremos al final de ella. En la pantalla de fin de receta podremos ver:

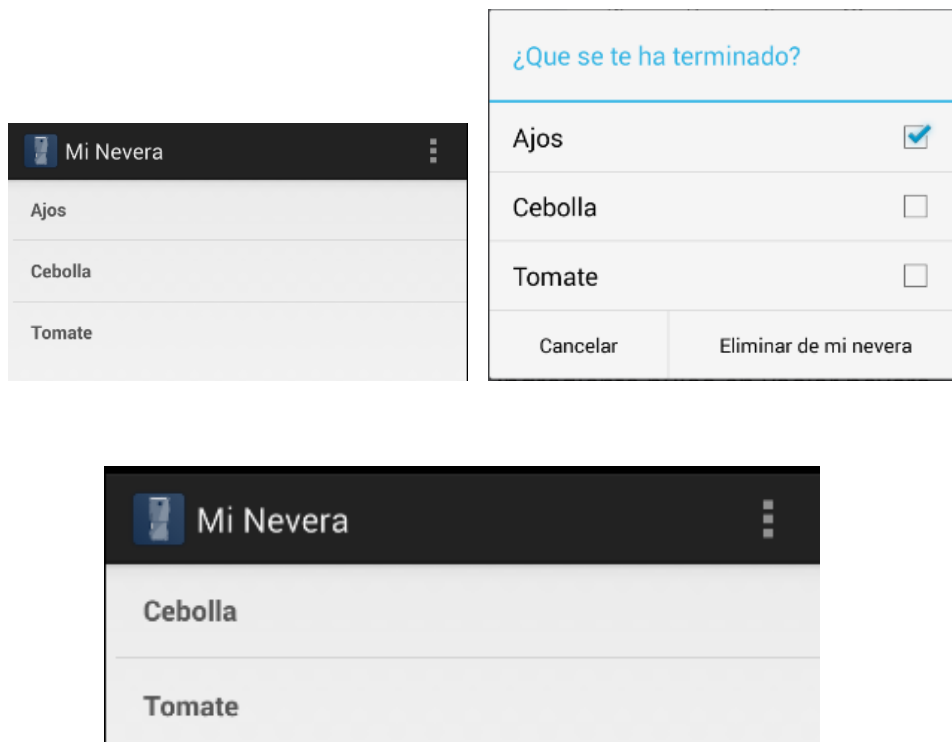


1. *Texto e imagen de fin de receta:* Para mostrar que la receta se ha terminado, se muestra una imagen y un texto que lo explica. Además se avisa al usuario de que en el caso de que se le haya acabado algún ingrediente, pulse en vaciar nevera.
2. *Botón anterior:* De la misma manera que desde cualquier paso se puede ir adelante o atrás, desde el final de la receta, el usuario puede volver a los pasos anteriores.
3. *Botón vaciar nevera:* En el caso en que, al realizar la receta, el usuario se haya quedado sin algún ingrediente, puede actualizar Mi Nevera pulsando Vaciar nevera.

Una vez pulsado el botón, la aplicación comprobará si hay algún ingrediente de la receta que esté dentro de Mi Nevera. En el caso en que los haya, se mostrará un listado de todos los disponibles:



El usuario puede seleccionar tantos ingredientes como se le hayan terminado. Una vez seleccionados, debe pulsar *Eliminar de mi nevera*. Una vez pulsado, la aplicación mostrará un mensaje avisando que se han eliminado los ingredientes correctamente.

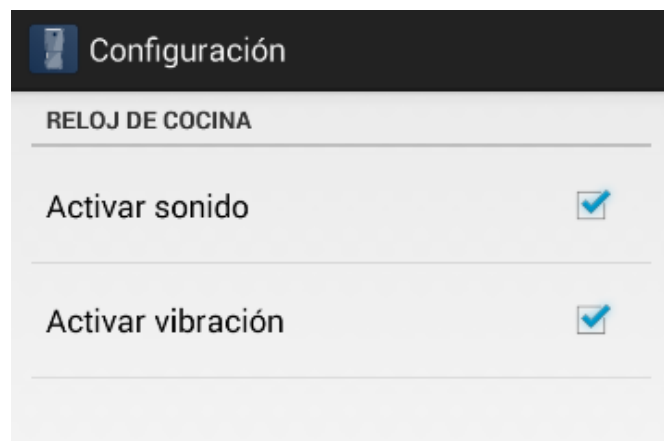


Opciones

En el último apartado de la aplicación podremos configurar la manera en que queremos que el reloj de cocina nos avise de que el tiempo se ha terminado.

Para acceder, sólo tenemos que pulsar en el botón Opciones desde la página principal.

Una vez dentro podremos ver la siguiente pantalla:



Desde aquí, podremos configurar si queremos que el reloj de cocina nos avise mediante sonido y/o vibración.

Estas opciones vienen marcadas por defecto.

Una vez se marquen o se desmarquen, estas quedarán guardadas automáticamente.

11. Conclusiones

Una vez finalizado el proyecto, y, después de muchas horas de trabajo, es la hora de exponer las conclusiones obtenidas en cuanto a varios aspectos de la realización de este.

Primero veremos las conclusiones obtenidas a la hora de evaluar los objetivos planteados inicialmente y, para finalizar expondré mis conclusiones personales sobre la realización del proyecto.

11.1 Evaluación de objetivos

- *Estudiar y aprender cómo funciona Android:* Al inicio del proyecto, desconocía por completo el desarrollo en Android. Meses después puedo decir que estoy muy lejos de ser una experta, pero si que he sido capaz de aprender lo básico para realizar una aplicación sencilla y, lo más importante, de poder aprender una nueva tecnología por mi misma sin necesidad de realizar ningún curso y sin ninguna ayuda profesional.
- *Analizar, diseñar e implementar una aplicación desde cero:* Creo que a cualquier desarrollador al que se le pregunte, le resulta mucho más gratificante realizar una aplicación creada por el mismo en vez de únicamente seguir unas directrices que marca otra persona. En este aspecto me considero satisfecha al haber realizado todo el proceso del desarrollo del software paso a paso aunque haya sido más costoso que en el caso de tener un proyecto previamente definido por otra persona.
- *Revisar y comprobar que el funcionamiento de la aplicación satisface los requisitos pensados inicialmente:* De todos los requisitos propuestos inicialmente, he acabado desarrollando todos ellos. Una vez acabado el desarrollo y después del testeo (una vez más gracias a la gente que me ha prestado sus dispositivos para ello) es muy gratificante ver las reacciones tan positivas de la gente ante la aplicación.

11.2 Conclusiones personales

En cuanto a lo que me ha parecido la realización del proyecto personalmente, primero recalcar lo difícil y agotador que me ha resultado compaginar mi trabajo a jornada completa con la realización del proyecto.

Una vez terminado, también puedo decir que ha sido muy enriquecedor el aprender una nueva tecnología por mi misma que, además puede ser muy útil a la hora de buscar trabajo en empresas de nueva constitución.

También puedo decir que estoy muy satisfecha con el producto obtenido ya que pienso que, con algunas de las mejoras mencionadas en uno de los capítulos anteriores, podría ser una buena aplicación.

11. Bibliografía

- Wikipedia: <http://es.wikipedia.org/>
- Stack overflow: <http://stackoverflow.com/>
- Android Developer: <http://developer.android.com/>
- El androide libre: <http://www.elandroidelibre.com/>
- Sgoliver.net blog : <http://www.sgoliver.net/>
- Dipler: <http://www.dipler.org/>
- Androidety: <http://androideity.com/>
- Eclipse: <http://www.eclipse.org/edt/>
- Genymotion: <http://www.genymotion.com/>
- SQLite database browser: <http://sourceforge.net/projects/sqlitebrowser/>
- Balsamiq: <http://balsamiq.com/products/mockups/>

